

# **Struktur, Funktion und Verhalten dynamischer Modelle der Systembiologie**

**Formale Wissensrepräsentation als Grundlage für  
computergestützte Modellierung und Simulation**

DISSERTATION

zur Erlangung des akademischen Grades  
doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der Fakultät für Mathematik und Informatik  
der Friedrich-Schiller-Universität Jena

von Diplom-Informatiker CHRISTIAN KNÜPFER  
geboren am 24.05.1972 in Jena

Jena, den 24. Oktober 2014

**Gutachter:**

1. Prof. Dr. Clemens Beckstein (Friedrich-Schiller-Universität Jena)
2. Prof. Dr. Falk Schreiber (Monash University, Clayton, Australien)
3. PD Dr. Stefan Artmann (Leopoldina, Halle/Saale)

**Tag der öffentlichen Verteidigung: 19. Oktober 2015**

# Inhaltsverzeichnis

<b>Einleitung</b>	<b>7</b>
<b>I. Dynamische Bio-Modelle als codiertes Wissen</b>	<b>13</b>
<b>1. Dynamische Modelle in der Systembiologie</b>	<b>15</b>
1.1. Dynamische Modellierung . . . . .	15
1.1.1. Modell und Wirklichkeit . . . . .	17
1.1.2. Modell, Simulation und Verhalten . . . . .	19
1.1.3. Performanz und Kompetenz von dynamischen Modellen . . . . .	20
1.2. Bio-Modelle . . . . .	23
1.2.1. Charakteristika von Bio-Modellen . . . . .	25
1.2.2. Methoden der Modellierung und Simulation in der Systembiologie . . . . .	26
1.2.3. Fallbeispiel: Tysons Modelle des Zellzyklus . . . . .	32
1.3. Computergestützte Modellentwicklung in der Systembiologie . . . . .	41
1.3.1. Modellentwicklungszyklus . . . . .	42
1.3.2. Allgemeine Anforderungen für die Entwicklung von Bio-Modellen . . . . .	45
1.3.3. Teilaufgaben einer computergestützten Modellentwicklung . . . . .	49
Zusammenfassung . . . . .	52
<b>2. Wissensfacetten von Bio-Modellen</b>	<b>53</b>
2.1. Interpretation von Bio-Modellen . . . . .	54
2.1.1. Duale Interpretation von Bio-Modellen . . . . .	54
2.1.2. Globale Metainformationen . . . . .	55
2.1.3. Anmerkungen zur biologisch-physikalischen Interpretation . . . . .	57
2.2. Drei Betrachtungsweisen von Bio-Modellen . . . . .	59
2.2.1. Struktur eines Bio-Modells . . . . .	61
2.2.2. Funktion eines Bio-Modells . . . . .	67
2.2.3. Verhalten eines Bio-Modells . . . . .	71
2.3. Vergleich mit anderen Konzeptualisierungen . . . . .	75
2.3.1. Duale Interpretation von Bio-Modellen . . . . .	75
2.3.2. Drei Betrachtungsweisen von Bio-Modellen . . . . .	76
2.3.3. Biologische Bedeutung von Bio-Modellen . . . . .	78
Zusammenfassung . . . . .	80

<b>II. Wissensrepräsentation für die Modellierung und Simulation</b>	<b>83</b>
<b>3. Wissensrepräsentation</b>	<b>85</b>
3.1. Formale Darstellung von Wissen . . . . .	85
3.1.1. Logikbasierte Wissensrepräsentation . . . . .	86
3.1.2. Wissen und Daten . . . . .	89
3.1.3. Knowledge Engineering . . . . .	91
3.2. Ontologien zum <i>Grounding</i> von Symbolen . . . . .	92
3.2.1. Das <i>Symbol Grounding Problem</i> . . . . .	93
3.2.2. Zum Ontologiebegriff . . . . .	96
3.2.3. Logische Präzision und ästhetische Prägnanz von Ontologien . . . . .	99
3.3. Informatische Ontologien . . . . .	100
3.3.1. Begriffsbestimmung informatischer Ontologie . . . . .	101
3.3.2. Spezifikation und Codierung informatischer Ontologien . . . . .	105
3.3.3. Entwicklung von Ontologien . . . . .	110
Zusammenfassung . . . . .	115
<b>4. Formale Repräsentation der Wissensfacetten von Bio-Modellen</b>	<b>117</b>
4.1. Bedingungen einer computergestützten Bio-Modellierung . . . . .	117
4.1.1. Infrastruktur für die Bio-Modellierung . . . . .	117
4.1.2. Minimale Informations-Standards . . . . .	122
4.1.3. Verbindung der intrinsischen und extrinsischen Wissensfacetten . . . . .	123
4.2. Formalisierung der Wissensfacetten von Bio-Modellen . . . . .	124
4.2.1. Formalisierung der Modell-Struktur . . . . .	125
4.2.2. Formalisierung der Modell-Funktion . . . . .	131
4.2.3. Formalisierung des Modell-Verhaltens . . . . .	136
4.3. Computergestützte Bio-Modellierung im wissenschaftlichen Kontext . . . . .	139
4.3.1. Formalisierung der globalen Metainformationen von Bio-Modellen . . . . .	139
4.3.2. Teleologische Funktion als Treiber für die Computerunterstützung . . . . .	141
4.3.3. Entwurf einer <i>Modelling Intention Ontology</i> (MINTENTO) . . . . .	142
Zusammenfassung . . . . .	146
<b>III. Dynamik von Bio-Modellen</b>	<b>149</b>
<b>5. Qualitative Beschreibung von Dynamik</b>	<b>151</b>
5.1. Theorie dynamischer Systeme . . . . .	151
5.1.1. Dynamische Systeme . . . . .	152
5.1.2. Zeitliches Verhalten dynamischer Systeme . . . . .	153
5.1.3. Qualitative Änderungen des Verhaltens: Bifurkationen . . . . .	157
5.2. Qualitative Modellierung und Simulation . . . . .	160
5.2.1. Qualitative Modelle . . . . .	161
5.2.2. Verwendung qualitativer Modelle . . . . .	165
5.2.3. Qualitatives Verhalten . . . . .	167
5.3. Temporale Logiken . . . . .	170
5.3.1. Modellierung dynamischer Systeme als Kripke-Struktur . . . . .	171

5.3.2. Spezifikation temporaler Eigenschaften . . . . .	175
5.3.3. Verifikation temporaler Eigenschaften . . . . .	177
Zusammenfassung . . . . .	182
<b>6. Terminologie dynamischen Verhaltens: TEDDY</b>	<b>185</b>
6.1. Inhalt von TEDDY . . . . .	185
6.1.1. Zeitliches Verhalten . . . . .	186
6.1.2. Diversifikation von Verhalten . . . . .	189
6.1.3. Funktionale Struktur-Muster . . . . .	193
6.2. Entwicklung von TEDDY . . . . .	194
6.2.1. Entstehungsgeschichte von TEDDY . . . . .	194
6.2.2. Implementierung von TEDDY . . . . .	195
6.2.3. Entwicklungsprozess am Beispiel der Stabilitätseigenschaften . . . . .	197
6.3. Verwendung von TEDDY . . . . .	201
6.3.1. Annotationen mit TEDDY . . . . .	201
6.3.2. DYML: Beschreibungssprache für die Dynamik von Bio-Modellen . . . . .	203
6.3.3. Repräsentation realen Verhaltens . . . . .	206
Zusammenfassung . . . . .	208
<b>Zusammenfassung und Ausblick</b>	<b>209</b>
<b>Literatur</b>	<b>213</b>
<b>Anhang</b>	<b>233</b>
<b>A. Abkürzungen und Projekte</b>	<b>235</b>
<b>B. ID-Verzeichnis</b>	<b>241</b>
<b>C. SBML-Kodierung von Modell 1</b>	<b>245</b>
<b>D. SBML-Kodierung von Modell 2</b>	<b>259</b>
<b>E. SED-ML Beispiel</b>	<b>267</b>
<b>F. TEDDY: Klassenhierarchie</b>	<b>269</b>
<b>Ehrenwörtliche Erklärung</b>	<b>273</b>



# Einleitung

„Eine der grundlegenden Aufgaben der Informatik besteht darin, die aus den Fachwissenschaften stammenden Modelle so umzuschreiben, daß sie mit Hilfe eines Computers dargestellt und bearbeitet werden können.“

---

(WEDEKIND ET AL. 1998: *Modellierung, Simulation, Visualisierung: Zu aktuellen Aufgaben der Informatik*, S. 265)

Die Systembiologie bietet große Chancen und steht vor ebenso großen Herausforderungen. Neue experimentelle Methoden versetzen uns in die Lage, lebende Systeme immer umfassender und immer detailreicher zu untersuchen. Auf Grundlage der dabei anfallenden experimentellen Daten werden zunehmend biochemische Prozesse mechanistisch modelliert und simuliert. Durch Integration solcher Teilmodelle kommt die Systembiologie ihrem Ziel stetig näher, lebende Systeme in ihrem inneren Wirken zu verstehen. Ein Meilenstein ist dabei etwa das im letzten Jahr veröffentlichte Modell **Recon 2** (THIELE ET AL. 2013), welches den gesamten menschlichen Metabolismus mathematisch rekonstruiert. Gleichzeitig stellt **Recon 2** eine Wissensbasis des menschlichen Metabolismus dar (<http://humanmetabolism.org>). Daran wird die enge Verbindung zwischen Modellen und Wissen deutlich, welche ein zentrales Thema der vorliegenden Arbeit ist. Die wachsende Anzahl und Kompliziertheit der Modelle und das immensen Wissens, welches notwendig ist, um ein Modell zu verstehen oder zu entwickeln, stellen besondere Herausforderungen für die Systembiologie dar. Diese Herausforderungen können nur mit Computerunterstützung bewältigt werden.

Der vorliegenden Arbeit dient das oben stehende Diktum von WEDEKIND ET AL. 1998 als Leitmotiv. Dieses Leitmotiv wird jedoch einerseits konkretisiert, indem sich die Arbeit auf die Modelle der Systembiologie fokussiert. Solche formalen, dynamischen Modelle, die zur Erklärung biologischer Phänomene eingesetzt werden, sind Hauptgegenstand der vorliegenden Arbeit und werden hier kurz als „Bio-Modelle“ bezeichnet. Andererseits wird das Leitmotiv insofern erweitert, als neben der Formalisierung der Modelle selbst auch die Formalisierung der Verwendung der Modelle in Simulationen und die Formalisierung der Ergebnisse dieser Simulationen betrachtet wird. Kurz gesagt wird es in dieser Arbeit um die *Struktur*, die *Funktion* und das *Verhalten* von Modellen in der Systembiologie gehen.

In den letzten Jahren haben sich formale Sprachstandards etabliert, in denen systembiologische Modelle codiert und zwischen entsprechenden Programmen ausgetauscht werden können. Obwohl es darüber hinaus vielversprechende Entwicklungen gibt, bestimmte für die Modellierung und Simulation relevante Wissensfragmente formal zu beschreiben, fehlte zu Beginn dieser Arbeit eine umfassende systematische Untersuchung der Prozesse der Modellierung und Simulation und des darin involvierten Wissens sowie eine entsprechende Einordnung existierender formaler Beschreibungen. Die in dieser Arbeit durchgeführte wis-

wissenschaftstheoretische Analyse der Modellierung und Simulation in der Systembiologie stellt eine bedeutende Grundlage für eine durchgehende wissensbasierte Computerunterstützung bei der Modellentwicklung dar, indem sie hilft, bestehende Ansätze zu bewerten und in Beziehung zu setzen sowie relevante Wissensbereiche zu identifizieren, für die es noch keine formalen Beschreibungsmittel gibt. Als Ergebnis der wissenschaftstheoretischen Analyse des Modellierungsprozesses werden Umfang und Gestalt der formalen Wissensrepräsentation als Grundlage für eine computergestützte Modellierung und Simulation bestimmt. Die zentrale Frage, die in dieser Arbeit beantwortet werden soll, lautet also:

*Welches Wissen muss in welcher Form formal repräsentiert sein, um als Grundlage für eine allumfassende computergestützte Modellierung und Simulation in der Systembiologie dienen zu können?*

Die zentrale Frage soll von zwei Seiten angegangen werden: theoretisch und praktisch. Theoretisch wird ausgehend von der wissenschaftstheoretischen Analyse der Modellierung und Simulation in der Systembiologie mit den „Wissensfacetten von Bio-Modellen“ ein konzeptuelles Schema entwickelt, welches das involvierte Wissen systematisiert. Dabei wird sich zeigen, dass Bio-Modelle (als formale Modelle) eine duale Natur haben: Sie können einerseits *intrinsisch* als formaler Ausdruck betrachtet werden und stehen andererseits für ein *extrinsisches* Phänomen. Darüber hinaus gibt es drei Betrachtungsweisen von Bio-Modellen (als dynamische Modelle): ihre Struktur, ihre Funktion und ihr Verhalten. Mit Hilfe der Wissensfacetten wird skizziert, wie eine formale Repräsentation als Grundlage für eine umfassende Computerunterstützung der Modellentwicklung erfolgen sollte. Dabei werden bestehende Ansätze eingeordnet und fehlende Sprachmittel identifiziert. Praktisch wird die zentrale Frage der vorliegenden Arbeit angegangen, indem mit [TEDDY](#) (*TErminology for the Description of DYnamics*, [COURTOT ET AL. 2011](#)) eine Ontologie zur Beschreibung der Dynamik von Bio-Modellen entwickelt und eine weitere Ontologie für die Funktion von Bio-Modellen konzipiert wird.

Dies ist keine systembiologische Arbeit. Es handelt sich stattdessen um eine originär informatische Arbeit. Dabei wird Informatik als Strukturwissenschaft, als Formalwissenschaft und als Ingenieurwissenschaft aufgefasst. Als Strukturwissenschaft analysiert die Informatik eine Klasse von gegebenen Objekten und versucht, Wiederkehrendes und Gleichförmiges – also Strukturen – auszumachen. Aufbauend auf dieser Analyse wird dann der Anwendungsbereich konzeptuell modelliert. Diese grundlegende informatische Tätigkeit der konzeptuellen Modellierung findet sich heute explizit in den Ontologien wieder. Jedem Programm liegt allerdings schon von jeher eine „Ontologie“, also ein konzeptuelles Modell zugrunde. Die Objekte in dieser Arbeit sind dynamische Modelle. Da es sich bei der computergestützten Modellierung um eine wissenschaftliche Methode handelt, wird die Analyse eine wissenschaftstheoretische sein. Als Ergebnis dieser Analyse wird in dieser Arbeit das konzeptuelle Schema der Wissensfacetten von Bio-Modellen entwickelt ([Teil I](#)). Als Formalwissenschaft überführt die Informatik das konzeptuelle Schema in eine formale Repräsentation ([Teil II](#)). Als Ingenieurwissenschaft schließlich setzt die Informatik die formalen Repräsentationen in funktionierende, praktisch nutzbare Artefakte um. In dieser Arbeit wird dargelegt, wie dynamisches Verhalten von Modellen computerverarbeitbar beschrieben und somit die systembiologische Modellierung durch den Computer unterstützt werden kann ([Teil III](#)). Dies ist somit eine informatische Arbeit, die für die Systembiologie hilfreich ist.



---

Die Arbeit ist in drei Teile gegliedert: einen ontologischen, einen normativen und einen konstruktiven. Im ontologischen [Teil I](#) wird das Wesen von Bio-Modellen als wissenschaftliche Objekte der Systembiologie untersucht. Im normativen [Teil II](#) wird dargestellt, in welcher Weise und welchem Umfang Wissen für eine computergestützte Modellierung und Simulation formal repräsentiert werden muss. Im konstruktiven [Teil III](#) schließlich wird mit [TEDDY](#) eine eigene Entwicklung vorgestellt, mit deren Hilfe sich die Dynamik von Bio-Modellen formal beschreiben lässt. Wie im Folgenden dargestellt, untersucht jeder Teil eine eigene These.

Der [Teil I](#) („**Dynamische Bio-Modelle als codiertes Wissen**“) verfolgt die These:

*Eine vollständige Beschreibung eines Bio-Modells muss seine Struktur, seine Funktion und sein Verhalten umfassen und die formale Semantik mit der biologischen Wirklichkeit in Beziehung setzen.*

Dafür wird sich zunächst in [Kapitel 1](#) („Dynamische Modelle in der Systembiologie“) den Bio-Modellen als Untersuchungsgegenstand der Arbeit zugewendet und eine computergestützte Modellentwicklung in der Systembiologie motiviert. In [Kapitel 2](#) („Wissensfacetten von Bio-Modellen“) wird analysiert, wie Bio-Modelle interpretiert und verwendet werden. Ausgehend von dieser Analyse werden die Wissensfacetten von Bio-Modellen entwickelt und abschließend mit verwandten Konzeptualisierungen verglichen.

In [Teil II](#) („**Wissensrepräsentation für die Modellierung und Simulation**“) wird folgende These verfolgt:

*Eine formale Repräsentation der Wissensfacetten von Bio-Modellen bietet eine Grundlage für computergestützte Modellierung und Simulation.*

In [Kapitel 3](#) („Wissensrepräsentation“) werden zunächst relevante Methoden vorgestellt, mit denen Wissen formal repräsentiert und durch den Computer verarbeitet werden kann. [Kapitel 4](#) („Formale Repräsentation der Wissensfacetten von Bio-Modellen“) widmet sich der formalen Darstellung der Wissensfacetten von Bio-Modellen. Dabei werden existierende Ansätze präsentiert und bewertet sowie fehlende Stellen identifiziert.

Der letzte [Teil III](#) („**Dynamik von Bio-Modellen**“) widmet sich der These:

*Das Verhalten von Bio-Modellen lässt sich auf der Grundlage einer Ontologie der Theorie dynamischer Systeme propositional formalisieren.*

Dafür werden zunächst in [Kapitel 5](#) („Qualitative Beschreibung von Dynamik“) drei wichtige Methoden vorgestellt, Verhalten qualitativ zu beschreiben: die Theorie dynamischer Systeme, die qualitative Modellierung und Simulation sowie temporale Logiken. In [Kapitel 6](#) („Terminologie dynamischen Verhaltens: TEDDY“) wird die Ontologie [TEDDY](#) entwickelt, mit deren Hilfe das Verhalten von Bio-Modellen propositional beschrieben werden kann.

In der vorliegenden Arbeit werden durchgehend zwei Beispiele von Bio-Modellen zur Illustration verwendet. Diese Beispiel-Modelle werden in [Abschnitt 1.2.3](#) vorgestellt. In [Anhang C](#) und [Anhang D](#) finden sich die vollständigen Codierungen beider Modelle in [SBML](#) (*Systems Biology Markup Language*, [HUCKA ET AL. 2003](#)). Alle in der Arbeit verwendeten Akronyme, sowie alle referenzierten Standards, formalen Sprachen, Ontologien, Werkzeuge und Projekte sind in [Anhang A](#) aufgeführt zusammen mit entsprechenden Referenzen

und Webseiten. In [Anhang B](#) sind alle in der Arbeit referenzierten Datenbankeinträge und Ontologietерme zusammen mit ihren zugehörigen [URIs](#) (*Uniform Resource Identifier*) aufgelistet. Schließlich enthält [Anhang F](#) die vollständige Klassenhierarchie von [TEDDY](#).

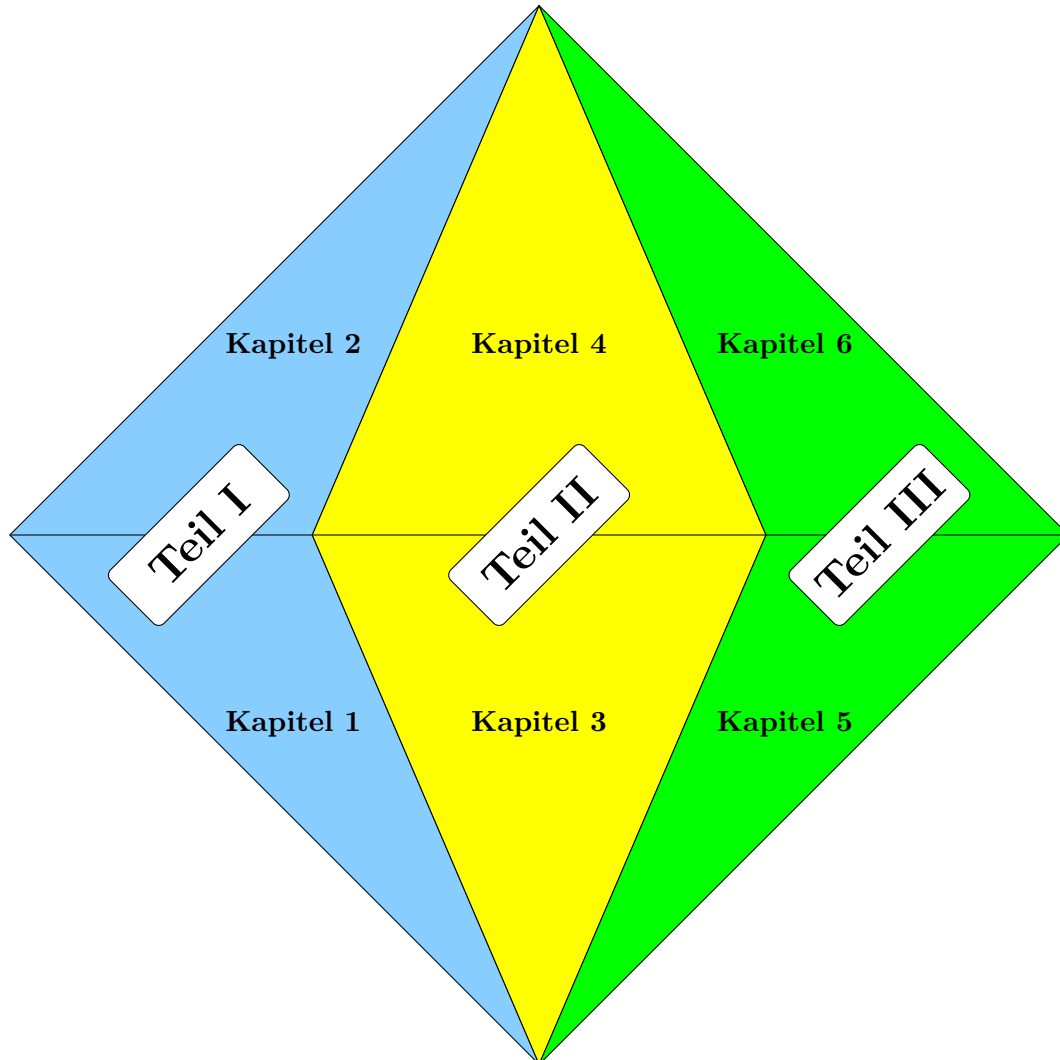


Abbildung 0.1.: Aufbau der vorliegenden Arbeit in Analogie zum „Wissensdiamanten“ ([Abbildung 2.7](#)): Die Farben verkörpern die Struktur (blau), die Funktion (gelb) und das Verhalten (grün) von Bio-Modellen. Unten finden sich die (extrinsischen) Theoriekapitel, oben die (intrinsischen) Kapitel, die eigene Beiträge darstellen.

Der Aufbau der vorliegenden Arbeit weist eine gewisse Analogie zu den Wissensfacetten auf: [Teil I](#) untersucht die Struktur des Wissens, welches in die systembiologische Modellierung und Simulation involviert ist, [Teil II](#) untersucht die Funktion der Wissensrepräsentation als Grundlage für die computergestützte Modellierung und Simulation und [Teil III](#) beschäftigt sich mit dem Verhalten von Modellen und dessen Formalisierung. Jeder der drei Teile hat zunächst ein Theoriekapitel, welches Bestehendes analysiert, also in gewisser Weise extrinsisch nach außen gerichtet ist: [Kapitel 1](#) stellt systembiologische

---

Modelle als den Untersuchungsgegenstand der Arbeit vor, [Kapitel 3](#) legt die methodischen Grundlagen der Wissensrepräsentation und [Kapitel 5](#) beschäftigt sich mit Ansätzen zur qualitativen Beschreibung von Dynamik. Jedem dieser Kapitel folgt ein Kapitel, welches einen eigenen Beitrag darstellt, also in gewisser Weise intrinsisch (bezüglich dieser Arbeit) ist: [Kapitel 2](#) präsentiert die Wissenfacetten der Modellierung und Simulation, [Kapitel 4](#) widmet sich der formalen Repräsentation der Wissensfacetten von Bio-Modellen und in [Kapitel 6](#) wird eine Ontologie für die formale Beschreibung der Dynamik von Modellen entwickelt. Der Zusammenhang zwischen den Wissensfacetten und dem Aufbau dieser Arbeit ist in [Abbildung 0.1](#) als Analogie zum „Wissensdiamanten“ (siehe [Abbildung 2.7](#)) dargestellt.



# Teil I.

## Dynamische Bio-Modelle als codiertes Wissen

„Ein Modell eines technischen oder natürlichen Systems repräsentiert das Wissen über bestimmte Aspekte des Verhaltens dieses Systems in einer formalen Sprache.“

---

([STRUSS 2000](#): *Modellbasierte Systeme und qualitative Modellierung*, S. 433)

Im ersten Teil der Arbeit wird untersucht, wie mit Hilfe von dynamischen Modellen unser Wissen über biologische Prozesse codiert wird. Dafür werden zunächst Modelle, wie sie in der Systembiologie verwendet werden, charakterisiert und der Begriff „Bio-Modell“ etabliert ([Kapitel 1](#)). Bio-Modelle stellen den Untersuchungsgegenstand dieser Arbeit dar. In [Kapitel 2](#) wird untersucht, welche Arten von Wissen in die Modellierung und Simulation biologischer Systeme involviert sind und welche Beziehungen zwischen den einzelnen Wissensfragmenten bestehen. Als Ergebnis einer Analyse von Modellierungsprozessen in der Systembiologie wird das beteiligte Wissen in einem konzeptuellen Schema systematisiert: den „Wissensfacetten von Bio-Modellen“. Die Wissensfacetten umfassen die Struktur, die Funktion und das Verhalten von Bio-Modellen aus algorithmischer (intrinsischer) und biologischer (extrinsischer) Sicht.



# 1. Dynamische Modelle in der Systembiologie

„The central task of systems biology is (a) to comprehensively gather information from each of these distinct levels for individual biological systems and (b) to integrate these data to generate predictive mathematical models of the system.“

---

([IDEKER ET AL. 2001](#): *A new approach to decoding life: systems biology*, S. 345)

In diesem Kapitel werden Bio-Modelle als der zentrale Forschungsgegenstand dieser Arbeit eingeführt. Kurz gesagt sind Bio-Modelle dynamische Modelle, die in der Systembiologie verwendet werden, um das Verhalten biologischer Systeme zu erklären. Im [Abschnitt 1.1](#) wird zunächst allgemein beschrieben, wie dynamische Phänomene mathematisch modelliert und simuliert werden können. Anschließend werden Bio-Modelle als spezielle Klasse dynamischer Modelle definiert und ihre Verwendung in der Systembiologie beschrieben ([Abschnitt 1.2](#)). In der gesamten Arbeit werden zwei konkrete Bio-Modelle zur Illustration verwendet. Diese Beispiel-Modelle werden in [Abschnitt 1.2.3](#) vorgestellt. Der abschließende [Abschnitt 1.3](#) dient als Motivation für die gesamte Arbeit: Hier wird die Vision einer computergestützten Modellentwicklung in der Systembiologie präzisiert.

Dieses Kapitel dient insbesondere auch der Klärung wichtiger Begriffe, die in dieser Arbeit verwendet werden, wie z.B.: (allgemeines) Modell, formales Modell, dynamisches Modell, Modellimplementierung, Modellinstanz, Zielsystem, formales System, dynamisches System, Simulation, Struktur, Funktion, Verhalten, Performanz, Kompetenz und schlussendlich Bio-Modell. Den hier referenzierten Arbeiten, die sich mit dem Thema Modellierung beschäftigen, liegen oft unterschiedliche Auffassungen dieser Begriffe zugrunde. Einige der referenzierten Arbeiten betreiben eine explizite Begriffsklärung. Hier soll keine ausführliche Diskussion der verschiedenen Begriffsauffassungen erfolgen. Stattdessen wird die hier verwendete Sicht auf die Begriffe dargelegt und wenn nötig die Wahl begründet. Dabei greift das Begriffsverständnis – bis auf die neu eingeführten Begriffe Performanz, Kompetenz und Bio-Modell – auf existierende Ansätze zurück. Es werden somit nicht unnötig neue Begriffsauffassungen eingeführt.

## 1.1. Dynamische Modellierung

In diesem Abschnitt wird die mathematische Modellierung natürlicher Phänomene aus wissenschaftstheoretischer Sicht betrachtet. Ausgehend vom allgemeinen Modellbegriff wird die spezielle Klasse der dynamischen Modelle eingeführt und charakterisiert.

Modelle sind allgegenwärtig in den Wissenschaften, begegnen uns aber auch im alltägli-

chen Leben. Sie sind wichtige Werkzeuge in den empirischen Wissenschaften (Erklärungs- und Simulationsmodelle), in der Mathematik und Logik (Modelltheorie), in der Informatik (Daten- und Verarbeitungsmodelle), in der Architektur (Baumodell), in Kunst und Mode (Fotomodelle, Mannequins), in der Didaktik (Planeten- und Atommodelle) und in der Freizeitgestaltung (Modelleisenbahn). Diese Aufzählung ließe sich sicherlich fortführen, sie soll jedoch lediglich demonstrieren, wie vielfältig Modelle verwendet werden und wie breit der Modellbegriff ist. All diesen Modellen ist gemeinsam, dass sie Modell *von* etwas sind. Damit ein Modell nützlich ist, muss diese Abbildung von etwas sich beschränken auf die relevanten Aspekte des Modellierten. Ein Modell ist also eine Abstraktion, Vereinfachung, Verkürzung. [STACHOWIAK 1973](#) identifiziert neben dem Abbildungs- und Verkürzungsmerkmal ein drittes Hauptmerkmal des allgemeinen Modellbegriffs: das pragmatische Merkmal. Dieses beschreibt „für wen, wann und wozu“ ([STACHOWIAK 1973](#), S. 133) etwas Modell ist.

Modelle repräsentieren also etwas in vereinfachter Form für einen bestimmten Zweck. Eine gewisse Abweichung von dieser allgemeinen Charakterisierung des Modellbegriffs stellen die Modelle der Modelltheorie dar, mit deren Hilfe die Semantik von logischen Theorien formalisiert wird. Hier dreht sich die Repräsentationsrelation um: Während z.B. ein mathematisches Pendel-Modell reale Pendel repräsentiert und deren Verhalten beschreibt, repräsentiert eine logische Theorie all ihre Modelle. [WEDEKIND ET AL. 1998](#) (S. 266) nennt diese Interpretationen logischer Theorien auch „Strukturmodelle“, da Theorien, die dieselben Modelle besitzen, strukturell identisch sind. [SUPPES 1960](#) versucht, Modelle im nicht-logischen Sinne dem seiner Meinung nach fundamentalen Modellbegriff der Semantik logischer Theorien unterzuordnen. Dafür fasst er logische Modelle als nichtsprachliche, erfüllende Realisierungen sprachlicher Theorien auf. Modelle im nicht-logischen Sinne, wie etwa physikalische Modelle, definieren dann lediglich die Menge der Objekte zugehöriger logischer Modelle ([SUPPES 1960](#), S. 290f). Diese Auffassung, die bereits von [STACHOWIAK 1973](#) (S. 4) kritisiert wurde, ignoriert den Fakt, dass Modelle oftmals als Theorien verwendet werden und durchaus sprachlich ausgedrückt sind, wie es z.B. von [KUIPERS](#) formuliert wird:

„A model is a (small) finite description of an infinitely complex reality, constructed for the purpose of answering particular questions.“ ([KUIPERS 1994](#), S. 2)

Dies gilt insbesondere für die dynamischen Modelle in der Systembiologie, die Hauptgegenstand dieser Arbeit sind. Im Folgenden sollen also Modelle als Repräsentationen der Wirklichkeit angesehen werden. Eine geeignete Interpretation eines Modell stellt den Bezug zur modellierten Wirklichkeit her. Im Gegensatz dazu sind die Modelle der Modelltheorie selbst schon Interpretationen entsprechender Theorien und diese Theorien stellen Repräsentationen der Wirklichkeit dar.

Für diese Arbeit ist es nicht notwendig, den allgemeinen Modellbegriff weiter zu diskutieren. Eine umfassende Charakterisierung verschiedenster Modelle findet sich in [STACHOWIAK 1973](#). [MAHR 2009](#) bietet eine aktuelle Diskussion unterschiedlicher Modellbegriffe aus Sicht der Informatik. Die speziellen Modelle, um die es im Folgenden gehen soll, sind formale Modelle, d.h. Beschreibungen in einer formalen Sprache, die relevante Aspekte des modellierten Wirklichkeitsausschnitts repräsentieren. Im folgenden Abschnitt werden zunächst die Beziehungen zwischen einem formalen Modell und der modellierten Wirklichkeit betrachtet ([Abschnitt 1.1.1](#)). Anschließend wird gezeigt, wie dynamische Modelle in Simulationen



eingesetzt werden, um das zeitliche Verhalten der modellierten Systeme zu untersuchen ([Abschnitt 1.1.2](#)). Schließlich werden zwei wichtige Eigenschaften – Performanz und Kompetenz – dynamischer Modelle eingeführt, die für die spätere Definition von Bio-Modellen notwendig sind ([Abschnitt 1.1.3](#)).

### 1.1.1. Modell und Wirklichkeit

Ein formales Modell ist eine Repräsentation eines Ausschnitts der Wirklichkeit. Bewusst wird an dieser Stelle der Begriff „Wirklichkeit“ verwendet und nicht der geläufigere Begriff „Realität“. In Abgrenzung vom metaphysischen Realismus wird hier davon ausgegangen, dass ein Zugang zur objektiven Realität nur über eine gegebene Konzeptualisierung erfolgen kann. Unsere (subjektive) Wahrnehmung der Realität ist also immer schon „theorie-beladen“ ([MOULINES 1994](#)). Die *Wirklichkeit* (englisch „actuality“) ist dann – in Übereinstimmung mit [PESCHL ET AL. 1999](#) – eine von einer gewählten Konzeptualisierung vermittelte Sicht auf die Realität (englisch „reality“). Dieser hier eingenommene Standpunkt ist in gewisser Weise *konstruktivistisch*. An dieser Stelle ist es nicht notwendig, auf die mit diesem Thema verbundenen philosophischen Diskussionen näher einzugehen.

Für die folgenden Ausführungen ist es nützlich, den modellierten Ausschnitt der Wirklichkeit als *System* im Sinne der Systemtheorie aufzufassen. Da formale Modelle selbst (formale) Systeme etablieren, werden hier folgende Begriffe unterschieden: Das *Zielsystem* ist der als System aufgefasste Ausschnitt der Wirklichkeit. Das *formale System* ist das durch das Modell spezifizierte System.

Wie verhält sich ein Modell zur modellierten Wirklichkeit? Während der *Modellbildung* müssen die relevanten Merkmale des modellierten Wirklichkeitsausschnitts im Modell abgebildet werden. Das durch das Modell spezifizierte formale System *codiert* somit das Zielsystem. Die durch Verwendung des formalen Systems (siehe [Abschnitt 1.1.2](#)) gewonnenen Resultate gestatten, wenn sie geeignet reinterpretiert (*decodiert*) werden, Aussagen über das Zielsystem. [ROSEN 1991](#) bezeichnet diese Beziehung zwischen formalem System und Zielsystem als „*Modeling Relation*“ (vgl. [Abbildung 1.1](#)): Die *Modeling Relation* begründet eine Kongruenzrelation zwischen den Schlussfolgerungen („inference“) im formalen System und den kausalen Zusammenhängen („causal entailment“) zwischen Beobachtungen des Zielsystems. Bei [ROSEN 1991](#) wird das Zielsystem als „natürliches System“ bezeichnet. Da jedoch auch artifizielle Systeme (z.B. Verbrennungsmotoren) modelliert werden können, scheint der Begriff „natürliches System“ unangemessen. Hier wird – in Anlehnung an [WEBB 2001](#) – stattdessen der Begriff „Zielsystem“ („target system“) verwendet. Bei den Modellen der Systembiologie sind die Zielsysteme biologische Natur, weshalb in der vorliegenden Arbeit oft auch der spezifischere Begriff „biologisches System“ verwendet wird.

Für [MORRISON ET AL. 1999](#) repräsentiert ein Modell nicht nur den modellierten Wirklichkeitsausschnitt sondern auch die wissenschaftliche Theorie, in dessen Rahmen es entwickelt wurde. In diesem Sinne *vermitteln* („mediate“) Modelle zwischen Theorie und Wirklichkeit:

„It is because they [models – Anmerkung des Autors] are neither one thing nor the other, neither just theory nor data, but typically involve some of both (and often additional ‚outside‘ elements), that they can mediate between theory and the world.“ ([MORRISON ET AL. 1999](#), S. 10f)

Das Modellsein kommt einem Artefakt nicht per se zu. Die Eigenschaft „Modellsein“ ist

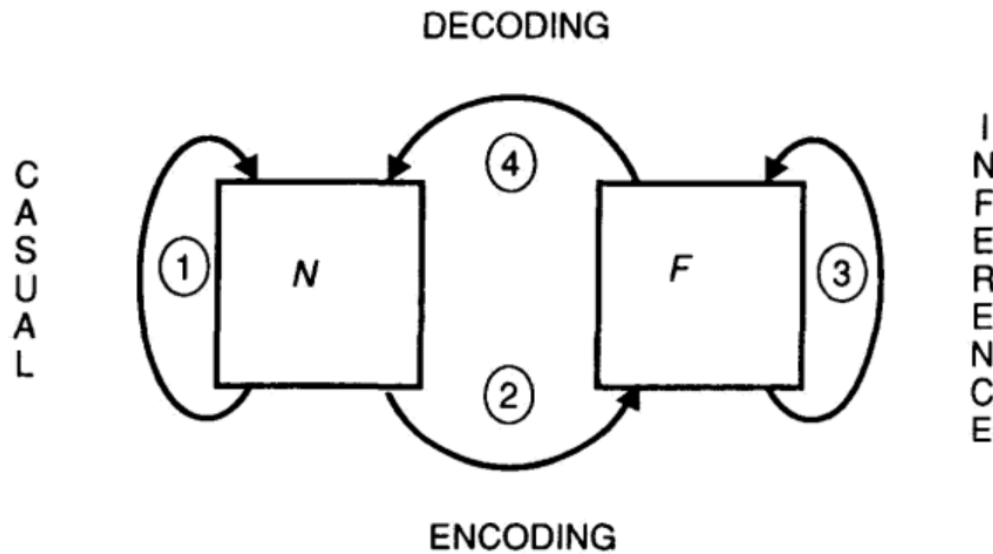


Abbildung 1.1.: *Modeling Relation* (aus ROSEN 1991, S. 60, Figure 3H.2)

keine immanente Eigenschaft eines formalen Ausdrucks. Stattdessen ist „von etwas Modell sein“ immer eine Zuschreibung oder, wie MAHR es nennt, eine „Auffassung“:

„Die Beziehung von Modell und Realität ist im hier beschriebenen Modell des Modellseins durch das Konzept der Auffassung bestimmt. Ein konkreter oder abstrakter Gegenstand (M) wird als Modell *aufgefasst*, d.h. es gibt eine intentionale Beziehung zwischen dem urteilenden Subjekt und M.“ (MAHR 2009, S. 247)

So ist es durchaus möglich, dass ein und derselbe formale Ausdruck verschiedene Wirklichkeitsausschnitte repräsentiert, also auch verschiedene Modelle verkörpert. Beispielsweise kann die Differentialgleichung  $\dot{x} = \alpha x, \alpha > 0$  unter anderem Modell für die Ausbreitung von Viren und für die nukleare Kernspaltung sein.

Der Zugang zur durch ein Modell repräsentierten Wirklichkeit erfolgt durch *Beobachtungen*. Diese Beobachtungen stammen im wissenschaftlichen Kontext meist aus *Experimenten* und liegen in Form von *Daten* vor. Bei dynamischen Modellen betreffen diese Beobachtungen *zeitliche Phänomene*.

Noch einmal in Kürze: Ein formales Modell eines Wirklichkeitsausschnitts spezifiziert ein formales System, welches das zum Wirklichkeitsausschnitt gehörende Zielsystem repräsentiert. Die Repräsentation des Zielsystems durch das formale System geschieht durch geeignete Codierung relevanter Merkmale des Zielsystems im formalen System und gestattet die Decodierung von mit dem formalen System gewonnenen Resultaten in Aussagen über das Zielsystem.

### 1.1.2. Modell, Simulation und Verhalten

Im letzten Abschnitt wurde gezeigt, wie formale Modelle Wissen über die Wirklichkeit codieren. *Dynamische Modelle* sind eine bedeutende Teilklasse der formalen Modelle, die zeitliche Veränderungen repräsentieren. In diesem Abschnitt wird beschrieben, wie dynamische Modelle in Simulationen verwendet werden, um Verhalten zu generieren. Dieses Verhalten kann zu Aussagen über das Zielsystem decodiert werden. Dadurch ist es unter bestimmten Voraussetzungen (siehe [Abschnitt 1.1.3](#)) möglich, beobachtete zeitliche Phänomene des modellierten Wirklichkeitsausschnitts mit Hilfe des Modells mechanistisch zu erklären. Manchmal findet sich auch der Begriff „Simulationsmodell“ (z.B. [WEDEKIND ET AL. 1998](#)), der die Verwendung des Modells in den Mittelpunkt stellt. Hier wird der allgemeinere Begriff des dynamischen Modells bevorzugt, um deutlich zwischen einem Modell und seiner Verwendung zu trennen.

Die Simulation dynamischer Modelle wird üblicherweise mit Hilfe von Computerprogrammen (*Simulatoren*) durchgeführt. Dafür muss das Modell zunächst in einer geeigneten computerverarbeitbaren formalen Sprache implementiert werden, z.B. in [SBML](#) (*Systems Biology Markup Language*, [HUCKA ET AL. 2003](#)). Ein so *implementiertes Modell* wird manchmal als „computational model“ bezeichnet (z.B. in [CHELLIAH ET AL. 2009](#)). Diese Bezeichnung ist jedoch irreführend, da nicht das Modell „computational“, d.h. berechnend bzw. computergestützt ist, sondern seine Verwendung. In dieser Arbeit wird stattdessen die Bezeichnung „implementiertes Modell“ verwendet.

In der Regel beschreibt ein dynamisches Modell eine große Anzahl verschiedener Verhaltensweisen. Erst durch die Benutzung in geeigneten Simulationen unter plausiblen Rand- und Anfangsbedingungen werden die Verhalten generiert, die in Aussagen über die modellierten zeitlichen Phänomene decodiert werden können. Zur vollständigen Beschreibung eines dynamischen Modells gehören also auch Informationen über die intendierte Verwendung des Modells. Das formale Modell selbst, d.h. die Spezifikation des formalen Systems, wird im folgenden die *Struktur* des Modells genannt. Seine intendierte Verwendung wird als *Funktion* bezeichnet. Zur Funktion gehört insbesondere auch die Angabe des Zielsystems. Das durch Simulationen des Modells gewonnene *Verhalten* komplettiert die Beschreibung eines dynamischen Modells.

Wie bereits dargelegt, spezifizieren formale Modelle formale Systeme. Im Falle dynamischer Modelle können in den spezifizierten formalen Systemen oft zwei Darstellungsweisen unterschieden werden: (1) Das formale System kann als *Netzwerk* dargestellt werden, d.h. als Menge von Entitäten und Relationen zwischen diesen. (2) Das formale System besitzt einen Zustand und verfügt über mathematische Beschreibungen der zeitlichen Veränderungen des Systemzustands. Die mathematische Darstellung des formalen Systems wird als *dynamisches System* bezeichnet, entsprechend der Begriffsverwendung in der Theorie dynamischer Systeme (siehe [Abschnitt 5.1](#)).

Ein dynamisches Modell enthält in der Regel formale Parameter, die es ermöglichen, mit Hilfe des Modells Verhalten bei unterschiedlichen Randbedingungen zu simulieren. Aus demselben Grund ist der Anfangszustand des formalen Systems im Modell in der Regel nicht festgelegt. Für die Verwendung des Modells in einer Simulation unter konkreten Rand- und Anfangsbedingungen müssen die formalen Parameter durch entsprechende Werte ersetzt (aktuelle Parameter) und der Anfangszustand des formalen Systems festgesetzt werden. Ein um diese Information erweitertes dynamisches Modell soll als *Modellinstanz*

oder instantiiertes Modell bezeichnet werden.

Das Verhalten eines Modells lässt sich sowohl quantitativ als auch qualitativ beschreiben. Die Simulation eines Modells liefert zunächst quantitative Resultate. Diese Resultate können bezüglich eines geeigneten Begriffssystems qualitativ interpretiert werden, d.h. sie werden propositional mit Hilfe der durch das Begriffssystem gegebenen Prädikate reformuliert. Eine qualitative Beschreibung des Modellverhaltens soll hier als *Dynamik* bezeichnet werden.

Der Fokus dieser Arbeit liegt auf dynamischen Modellen und deren Verwendung in Simulationen. Unter einer *einfachen Simulation* wird hierbei eine Berechnung des zeitlichen Verhaltens des Modells unter konkreten Rand- und Anfangsbedingungen verstanden. Eine *kombinierte Simulation* besteht aus mehreren einfachen Simulationen unter verschiedenen Rand- und Anfangsbedingungen und verbindet die Resultate der einfachen Simulationen zu einem Gesamtergebnis. Mit *Simulation* ist entweder eine einfache oder eine kombinierte Simulation gemeint.

Es gibt andere Verfahren, die es gestatten, Aussagen über das Verhalten dynamischer Modelle zu treffen, wie z.B. analytische Lösungen von Differentialgleichungssystemen, *Flux Balance Analysis* (ORTH ET AL. 2010) und *Chemical Organisation Theory* (DITTRICH ET AL. 2007). Verfahren, die nicht auf Simulationen des zeitlichen Verhaltens zurückgreifen, sind nicht Gegenstand dieser Arbeit.

An dieser Stelle sei darauf hingewiesen, dass es zwei unterschiedliche Aspekte der Dynamik von Modellen gibt: Ein Modell kann zum einen das dynamische Verhalten eines Zielsystems beschreiben (Dynamik-Modell oder dynamisches Modell), zum anderen unterliegt das Modell selbst im wissenschaftlichen Prozess Veränderungen (Modell-Dynamik). Dieser zweite Aspekt – die dynamische Entwicklung eines Modells – könnte durch ein dynamisches (Meta-)Modell beschrieben werden. Dabei wird das betrachtete Modell selbst zum Zielsystem.

Im folgenden Abschnitt wird gezeigt, wie die Simulation dynamischer Modelle für die Erklärung zeitlicher Phänomene eingesetzt werden kann. Dabei spielen die Performanz und die Kompetenz dynamischer Modelle eine zentrale Rolle. [Abbildung 1.8](#) am Ende dieses Kapitels fasst die wichtigsten Begriffe dynamischer Modelle und deren Simulation noch einmal zusammen.

### 1.1.3. Performanz und Kompetenz von dynamischen Modellen

Eines der Ziele dynamischer Modellierung ist die Gewinnung *mechanistischer Erklärungen* zeitlicher Phänomene. Mechanistische Erklärungen begründen die Entstehung des zu erklärenden Phänomens mit Hilfe von Mechanismen, wobei

„Mechanisms are composed of both entities (with their properties) and activities. Activities are the producers of change. Entities are the things that engage in activities.“ (MACHAMER ET AL. 2000, S. 3)

Das bedeutet, eine mechanistische Erklärung zeitlicher Phänomene basiert auf Objekten („entities“) und Aktivitäten dieser Objekte.

Bezogen auf das formale System, das durch ein dynamisches Modell spezifiziert wird, dürfen die darin vorkommenden Entitäten und Relationen jedoch nicht willkürlich sein. Damit das Modell für eine mechanistische Erklärung dienen kann, müssen die Entitäten und Relationen des formalen Systems strukturerhaltend auf entsprechende Objekte und

Aktivitäten des Zielsystems abgebildet werden können. Ein Modell heißt *kompetent* bezüglich des Zielsystems, wenn es eine strukturerhaltende Abbildung des formalen Systems auf das Zielsystem gibt. Kompetente Modelle spiegeln also die Struktur des Zielsystems wider. Dabei ist zu beachten, dass die Struktur des Zielsystems oder Teile davon meist nur in Form einer Hypothese bekannt sind. Dynamische Modelle dienen ja gerade der Aufstellung von Hypothesen über diese Struktur (vgl. [Abschnitt 1.2](#) und [Abschnitt 1.3.1](#)). Die Forderung der Kompetenz bezieht sich damit auf eine plausible Struktur des Zielsystems relativ zum aktuellen Wissensstand.

Ein kompetentes dynamisches Modell ist in der Lage, ein zeitliches Phänomen mechanistisch zu erklären. Ob diese Erklärung gültig ist, hängt davon ab, ob das Modell fähig ist, das zeitliche Phänomen unter plausiblen Rand- und Anfangsbedingungen in einer Simulation zu *imitieren*. Imitation heißt dabei, dass die Simulationsresultate das zeitliche Phänomen hinreichend genau nachbilden. Was dabei als hinreichend genau angesehen wird, hängt vom Verwendungskontext der Simulation ab. Ein dynamisches Modell heißt *performant* bezüglich eines zeitlichen Phänomens, wenn es eine Simulation unter plausiblen Rand- und Anfangsbedingungen gibt, in der das zeitliche Phänomen imitiert wird. Während Kompetenz mit der Struktur eines Modells zusammenhängt, bezieht sich Performanz also auf das Verhalten von Modellinstanzen und somit zusätzlich auf die Modellfunktion (siehe [Abschnitt 1.1.2](#)).

In [Abschnitt 1.1.2](#) wurden zwei Darstellungsformen für ein durch ein dynamisches Modell spezifiziertes formales System unterschieden: Netzwerk und dynamisches System. Die für die Kompetenz notwendige strukturerhaltende Abbildung erfolgt meist vom Netzwerk auf das Zielsystem. Das dynamische System ist eine notwendige Voraussetzung für die Durchführung von Simulationen und damit für die Performanz des Modells.

[ZEIGLER ET AL. 2000](#) (S. 30f) fasst diese beiden Eigenschaften dynamischer Modelle als unterschiedliche Gültigkeitskriterien auf: Ein performantes Modell besitzt „replicative validity“ und ein kompetentes Modell „structural validity“. Zusätzlich kann ein Modell auch noch „predictive validity“ besitzen, also sinnvolle Vorhersagen gestatten. In [WEBB 2001](#) (S. 1043f) wird Kompetenz als „structural accuracy“ und Performanz als „match“ bezeichnet. Für [ROSEN 1991](#) ist Performanz sogar eine definierende Eigenschaft eines Modells (vgl. [Abbildung 1.1](#)):

„The formal system F is called a model of the natural system N if we always get the same answer, whether we follow path (1), or whether we follow path (2)+(3)+(4).“ (S. 61)

Ob hiermit auch die Kompetenz gemeint ist, hängt davon ab, welche Ansprüche an die Codierung (2) des Zielsystems gestellt werden.

Im Folgenden werden die prägnanteren Begriffe Kompetenz und Performanz bevorzugt, die natürlich auf CHOMSKY zurückgehen:

„We thus make a fundamental distinction between *competence* (the speaker-hearer’s knowledge of his language) and *performance* (the actual use of language in concrete situations).“ ([CHOMSKY 1965](#), S. 4)

In Sinne von [CHOMSKY 1965](#) könnte man das Wissen über ein zeitliches Phänomen (Kompetenz) von der Benutzung dieses Wissens zum Imitieren des zeitlichen Phänomens (Performanz) unterscheiden.

Performanz und Kompetenz sind eng verbunden mit der Unterscheidung zwischen der Chronologie und der Kausalität von Ereignissen. Ohne hier auf Details einzugehen, lässt sich Folgendes sagen: Performante dynamische Modelle sind in der Lage, *zeitliche* Abfolgen von Systemzuständen widerzuspiegeln. Kompetente Modelle bieten Erklärungen für zeitliche Phänomene, sind also in der Lage, Warum-Fragen zu beantworten. Kompetente Modelle repräsentieren damit *kausale* Zusammenhänge.

Wann ein Modell als performant oder kompetent angesehen wird, hängt in starkem Maße vom Verwendungskontext und der Zielstellung des Modellierers ab. Somit sind Performanz und Kompetenz zunächst subjektive Urteile über ein dynamisches Modell. Für einen festgelegten Verwendungskontext kann die Performanz durch ein geeignetes Ähnlichkeitsmaß zwischen den Simulationsresultaten und den experimentellen Ergebnissen objektiviert werden. Da ein Modell eine Abstraktion darstellt, ist die Abbildung vom formalen System auf das gesamte Zielsystem im Allgemeinen nicht surjektiv. Für die Objektivierung der Kompetenz müssen für das Zielsystem die relevanten strukturellen Größen festgelegt werden, auf die das formale System abbilden soll. Idealerweise sollte die Abbildung jedoch total sein, d.h. jedem Element und jeder Relation des formalen Systems sollten Entsprechungen im Zielsystem zugeordnet sein. Ob und in welchem Maße fehlende Zuordnungen vom formalen System auf das Zielsystem Mängel des Modells oder unvollständiges Wissen über den modellierten Wirklichkeitsausschnitt darstellen, kann nur im wissenschaftlichen Diskurs selbst beurteilt werden. Diese Beurteilung stellt eine der wichtigsten Leistungen bei der Modellierung dar. So drehen sich historische Auseinandersetzungen in der Physik häufig um die physikalische Interpretation hypothetischer Größen (z.B. Kraft) und Konstanten (z.B. Plancksches Wirkungsquantum).

Sowohl die Performanz als auch die Kompetenz von dynamischen Modellen sind graduelle Eigenschaften. Wenn sie geeignet objektiviert sind, können sie auch quantitativ erfasst werden. Die beiden graduellen Eigenschaften Performanz und Kompetenz sind unabhängig voneinander. Es lassen sich demnach vier Extremfälle vorstellen (vgl. [Abbildung 1.2](#)): (1) Ein in keiner Weise performantes und kompetentes Modell hat keinerlei Imitations- und Erklärungsfähigkeit. Eigentlich verdient es damit auch nicht die Bezeichnung „Modell“. Aufgrund der fehlenden Strukturzuordnung zum Zielsystem können auch Verbesserungsversuche nicht zielgerichtet erfolgen. (2) Ausschließlich performante Modelle, auch Blackbox-Modelle genannt, imitieren das zeitliche Verhalten des Zielsystems, können aber keine mechanistischen Erklärungen für dieses Verhalten liefern. Sie können aber dennoch für die Erklärung zeitlicher Phänomene eine wichtige Rolle spielen, wenn sie als Bausteine in umfassenderen Modellen eingesetzt werden. Dort repräsentieren sie lokale dynamische Aspekte des Gesamtsystems, deren mechanistische Entstehung entweder unbekannt ist oder für die Erklärung des globalen Verhaltens vernachlässigt werden kann. Im Sinne von [MACHAMER ET AL. 2000](#) stellen sie in diesem Erklärungskontext die fundamentalen Mechanismen dar. (3) Ein kompetentes Modell, welches nicht performant ist, weist entweder auf einen Fehler im Modell, in unserem Wissen über das Zielsystem oder in beidem hin. Im ersten Fall sind nicht alle relevanten Objekte und Aktivitäten des Zielsystems im Netzwerk repräsentiert oder das dynamische System beschreibt die zeitlichen Veränderungen unzureichend genau. Im zweiten Fall fehlen meist bestimmte Einflussgrößen, d.h. sowohl das Netzwerk als auch die Struktur des Zielsystems sind unvollständig. Dieser Fall tritt häufig am Beginn der Modellbildung ein. Durch schrittweises Ändern des Modells und Verbesserung der Auffassung des Zielsystems gelangt man zum



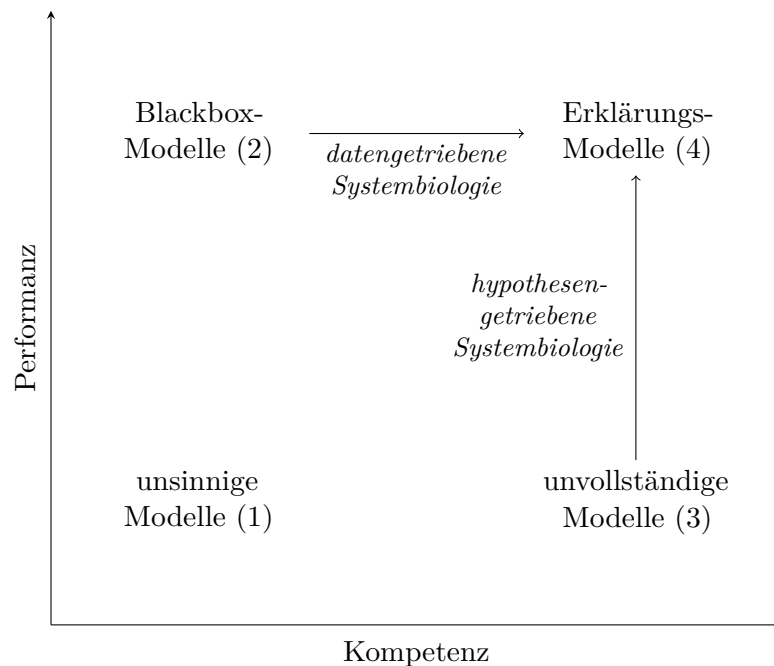


Abbildung 1.2.: Performanz und Kompetenz als graduelle Eigenschaften dynamischer Modelle. Je nach Ausprägung der beiden Eigenschaften können vier Arten von Modellen ausgemacht werden. Die Nummern in Klammern dienen der Referenzierung im Haupttext, für Details siehe dort.

fertigen Erklärungsmodell. (4) Das fertige Erklärungsmodell ist sowohl performant als auch kompetent. Es kann jetzt zur Prädiktion unbekannter Eigenschaften des modellierten Wirklichkeitsausschnitts eingesetzt werden. Diese Prädiktionen können im Rahmen neuer experimenteller Beobachtungen am Zielsystem verifiziert oder falsifiziert werden. Der interessantere Fall der Falsifikation führt dazu, dass das Modell nicht mehr performant (bezüglich der neu beobachteten zeitlichen Phänomene) ist. Damit wird eine neue Iteration im Modellierungszyklus (siehe [Abschnitt 1.3.1](#)) begonnen.

Im folgenden [Abschnitt 1.2](#) werden zwei Ansätze der Systembiologie beschrieben, deren grundlegende Methodik sich in [Abbildung 1.2](#) darstellen lässt: In der datengetriebenen Systembiologie werden aus experimentellen Daten generierte Blackbox-Modelle (2) als Ausgangspunkt für die Entwicklung von kompetenten Erklärungsmodellen (4) verwendet. In der hypothesengetriebenen Systembiologie werden mechanistische Modellfragmente (3) so zusammengesetzt und adaptiert, dass sie zu performanten Erklärungsmodellen bezüglich gegebener biologischer Phänomene (4) werden.

## 1.2. Bio-Modelle

Dieser Abschnitt widmet sich dem Begriff „Bio-Modell“. Bio-Modelle stellen den zentralen Untersuchungsgegenstand dieser Arbeit dar. Sie werden in der Systembiologie verwendet, um zeitliche biologische Phänomene mechanistisch zu erklären. Es sind also kompetente,

dynamische Modelle. Zunächst werden die wichtigsten Charakteristika von Bio-Modellen herausgearbeitet ([Abschnitt 1.2.1](#)). Anschließend werden wichtige Typen von Bio-Modellen vorgestellt ([Abschnitt 1.2.2](#)). Den Abschluss dieses Abschnitts bildet das Fallbeispiel ([Abschnitt 1.2.3](#)), welches in der gesamten Arbeit zur Illustration herangezogen wird.

Bevor der Begriff „Bio-Modell“ näher charakterisiert wird, sollen noch einige Bemerkungen über die Systembiologie und die Bedeutung der Modellierung innerhalb dieser Forschungsrichtung vorweggeschickt werden. Die Systembiologie ist ein Zweig der Biologie. Ziel der Systembiologie ist das Erklären des Verhaltens biologischer Systeme. Im Gegensatz zur Molekularbiologie und Biochemie, in denen die (lokalen) Wechselwirkungen zwischen einzelnen Komponenten biologischer Systeme im Detail untersucht werden, wird in der Systembiologie analysiert, wie (globales) Systemverhalten aus einer Vielzahl lokaler Wechselwirkungen entsteht. Die Systembiologie bezieht sich dabei sowohl terminologisch als auch methodisch stark auf die Systemtheorie.

Es gibt zwei unterschiedliche methodische Ansätze in der Systembiologie, die oft als „*Top-down*“ und „*Bottom-up*“ bezeichnet werden ([BRUGGEMAN ET AL. 2007](#), vgl.): Die *Top-down* Systembiologie analysiert große Mengen von Daten, wie sie bei modernen Hochdurchsatzexperimenten anfallen. In den Daten werden Muster und Korrelationen gesucht. Aus diesem Grund wird diese Art Systembiologie auch als „discovery science“ ([IDEKER ET AL. 2001](#)) bezeichnet. Es handelt sich um ein typisches *Reverse Engineering* Verfahren, bei welchem versucht wird, aus beobachtetem Verhalten Rückschlüsse auf die zugrundeliegende Struktur zu ziehen. Im Gegensatz dazu geht die *Bottom-up* Systembiologie von bekannten Wechselwirkungen aus und versucht, durch mathematische Modellierung das Systemverhalten vorherzusagen. Diese Art Systembiologie basiert also auf Hypothesen in Form von Verhaltensvorhersagen und wird somit auch „hypothesis-driven science“ genannt ([IDEKER ET AL. 2001](#)). Entsprechend kann man die *Top-down* Systembiologie als „datengetriebene“ Wissenschaft bezeichnen.

Es herrscht keine Einigkeit über den Begriff „Systembiologie“. Oft wird nur einer der beiden oben genannten Ansätze als Systembiologie angesehen. [LANDER 2004](#) verwendet für die *Top-down* Systembiologie sogar den Begriff „bioinformatics“ (S. 0712), da die Bioinformatik traditionell Verfahren für die Analyse großer Datenmengen bereitstellt. Um die Bedeutung der computergestützten Modellierung und Simulation von biologischen Systemen hervorzuheben, bezeichnet [LANDER 2004](#) die *Bottom-up* Systembiologie als „computational biology“ (S. 0712). Letztendlich wird das Verstehen des Verhaltens komplexer biologischer Systeme aber einer Integration beider Ansätze bedürfen:

„The integration of these two approaches, discovery and hypothesis-driven science, is one of the mandates of systems biology.“ ([IDEKER ET AL. 2001](#), S. 344)

Dynamische Modelle spielen in der Systembiologie eine ganz entscheidende, zentrale Rolle, wie bereits aus dem Diktum am Kapitel-Anfang (siehe [Seite 15](#)) hervorgeht. Dynamische Modelle bilden den methodischen Kern im systembiologischen Erkenntnisprozess und werden gar als Integratoren von Wissen angesehen, wie die folgenden beiden Zitate zeigen:

„At the core of systems biology lies the construction of models describing biological systems.“ ([FISHER ET AL. 2007](#), S. 1239)



„Instead, formal, quantitative models replace these static diagrams as integrators of knowledge, and serve as the centerpiece of the scientific modeling and simulation cycle.“ (COURTOT ET AL. 2011, S. 1)

Dabei liefern beide Ansätze der Systembiologie als Ergebnis Modelle. Die Modelle der *Top-down* Systembiologie sind dabei zunächst rein phänomenologisch, d.h. sie beschreiben nur die Beobachtungen:

„The models used in top-down systems biology are phenomenological – that is, they are not based on mechanisms and, mostly, they do not even employ knowledge about relationships between molecular components. It should be noted that correlations are mapped onto virtual mechanisms, which might well describe the correlations but, nonetheless, can still be far from reality.“ (BRUGGEMAN ET AL. 2007, S. 47)

Im Sinne von [Abschnitt 1.1.3](#) sind diese Modelle also performant aber nicht kompetent. Kompetente Modelle entwickelt dagegen die *Bottom-up* Systembiologie mit dem Ziel, mechanistische Erklärungen zu liefern:

„These models [used in bottom-up systems biology – Anmerkung des Autors] are mechanism-based rather than phenomenological.“ (BRUGGEMAN ET AL. 2007, S. 47)

Eben jene kompetenten dynamischen Modelle der (*Bottom-up*) Systembiologie sind der Forschungsgegenstand der vorliegenden Arbeit. Im nächsten Abschnitt wird der entsprechende Begriff des „Bio-Modells“ genauer gefasst und charakterisiert.

### 1.2.1. Charakteristika von Bio-Modellen

In diesem Abschnitt wird der Begriff „Bio-Modell“ definiert. Dabei werden die Charakteristika herausgearbeitet, die Bio-Modelle von anderen Arten biologischer Modelle abgrenzen. Bio-Modelle sind dynamische, kompetente Modelle biologischer Systeme. Sie werden in Simulationen verwendet, um zeitliche Phänomene der biologischen Systeme zu imitieren (Performanz) und mechanistisch zu erklären (Kompetenz). Ein *Bio-Modell* ist also:

- (1) ein biologisches Modell, d.h. es modelliert ein biologisches Zielsystem,
- (2) ein formales Modell, d.h. es spezifiziert ein formales System,
- (3) ein dynamisches Modell, d.h. es modelliert zeitliche Phänomene des Zielsystems mit Hilfe eines dynamischen Systems und
- (4) ein kompetentes Modell bezüglich der zeitlichen Phänomene des Zielsystems, d.h. es spezifiziert ein formales System, welches strukturerhaltend auf das Zielsystem abgebildet werden kann.

[Punkt 2](#) folgt direkt aus [Punkt 3](#) und der Definition eines dynamischen Modells. [Punkt 4](#) ist eine Voraussetzung für mechanistische Erklärungen der modellierten zeitlichen Phänomene. Die andere Voraussetzung der Performanz bezüglich der zu erklärenden zeitlichen

Phänomene wird nicht als definierende Eigenschaft von Bio-Modellen verwendet. Stattdessen ist die Performanz ein Qualitätsmerkmal eines Bio-Modells.

Es kann durchaus diskutiert werden, ob die Einbeziehung der Kompetenz ([Punkt 4](#)) in die Definition von Bio-Modellen sinnvoll ist. Ein möglicher Einwand wäre, dass Kompetenz stark kontextabhängig ist und kein objektives Kriterium darstellt. Jedoch ist bereits das Modellsein selbst, wie in [Abschnitt 1.1.1](#) dargelegt, eine Zuschreibung, d.h. abhängig von einem urteilenden Subjekt. Bei einem Modell muss, um es als Modell zu kennzeichnen, angegeben werden, wovon es Modell ist. Bei einem Bio-Modell muss zusätzlich beschrieben werden, wie der Zusammenhang zwischen formalem System und Zielsystem beschaffen ist, wodurch es also zu einem kompetenten Modell bezüglich des Zielsystems gemacht wird. Durch die Einbeziehung der Kompetenz in die Definition sollen hier Bio-Modelle, die als mechanistische Erklärungen dienen, von rein statistischen biologischen Modellen abgegrenzt werden, die experimentelle Daten lediglich beschreiben. Aus diesem Grund nennt [ELLNER ET AL. 2006](#) letztere Art von Modellen „descriptive“, da sie ein „quantitative summary of the observed relationships among a set of measured variables“ bieten, jedoch „nothing about why the variables are related the way they are“ (S. 2) aussagen. Im Gegensatz dazu, fordert [ELLNER ET AL. 2006](#) von, wie es dort heißt, dynamischen Modellen („dynamic models“), dass sie mechanistisch sind, d.h. dass sie „explicitly considering the processes that produce our observations“ (S. 3). Hier wird also die Kompetenz bereits bei dynamischen Modellen gefordert. Die Einbeziehung der Kompetenz in die Definition von Bio-Modellen entspricht dem Modellbegriff der *Bottom-up* Systembiologie bei [BRUGGEMAN ET AL. 2007](#) (S. 47) als „mechanism-based rather than phenomenological“ (vgl. Einleitung zu [Abschnitt 1.2](#)).

Zum Abschluss dieses Abschnitts werden Beispiele für Modelle angegeben, die keine Bio-Modelle sind, um die einzelnen definierenden Eigenschaften zu illustrieren:

Beispiel 1. Eine mathematische Beschreibung der Bewegung eines Pendels ist *kein biologisches* Modell. Es ist zwar ein kompetentes dynamisches Modell, hat aber kein biologisches System zum Gegenstand.

Beispiel 2. Ein anatomisches Skelett-Modell ist zwar ein biologisches, jedoch *kein formales Modell*, sondern eine physische Rekonstruktion echter Skelette.

Beispiel 3. Die als Zeichenkette dargestellte Aminosäuresequenz (Primärstruktur) eines Proteins ist *kein dynamisches* Modell, da sie keine zeitlichen Veränderungen beschreibt. Die Zeichenkette stellt jedoch ein formales Modell dar.

Beispiel 4. Das in [Abschnitt 1.1.1](#) angeführte dynamische Modell  $\dot{x} = \alpha x, \alpha > 0$  für die Ausbreitung von Viren ist *kein kompetentes* Modell, wenn es um die Erklärung der Ausbreitung unter der Bedingung systematisch durchgeführter Schutzimpfungen geht. Zwar kann eventuell  $\alpha$  so angepasst werden, dass das Modell vorliegende Daten approximiert, das Modell also performant bezüglich dieser Daten ist. Eine Erklärung des Einflusses der Schutzimpfungen auf die Ausbreitung ist mit dem Modell jedoch nicht möglich, da die Schutzimpfung nicht im Modell repräsentiert ist.

### 1.2.2. Methoden der Modellierung und Simulation in der Systembiologie

In diesem Abschnitt soll ein Überblick über systembiologische Methoden zur Modellierung und Simulation biologischer Systeme gegeben werden. In der Systembiologie kommt

eine große Vielfalt an Modellierungsmethoden zur Anwendung, die sich sowohl in der epistemischen Konzeptualisierung des Zielsystems, als auch in den zugrunde liegenden mathematischen Formalismen und in den zugehörigen Simulationsverfahren unterscheiden. Grundsätzlich wird dabei die Wahl des verwendeten Formalismus stark von der epistemischen Konzeptualisierung und der konkreten Fragestellung abhängen:

„The choice of a mathematical model or an algorithm to describe a biological object depends on the problem, the purpose, and the intention of the investigator.“ (KLIPP ET AL. 2009, S. 10)

Die epistemische Konzeptualisierung biologischer Wirklichkeit umfasst einen großen Bereich räumlicher und zeitlicher Auflösung: Biologische Systeme reichen von der submolekularen Ebene im Nanometerbereich bis hin zu Ökosystemen im Erdmaßstab. Die zeitliche Auflösung kann dabei von Molekylvibrationen im Femtosekundenbereich bis zu evolutionären Entwicklungen in Erdzeitaltern reichen (vgl. KLIPP ET AL. 2009, S. 3f). Typische elementare Objekte von biologischen Systemen sind entsprechend der biologischen Organisationshierarchie molekulare Bindungen, Moleküle, Zellbestandteile, Zellen, Gewebe, Organe, Organismen und Populationen.

Üblicherweise beschränkt sich die Systembiologie auf die Beschreibung molekularbiologischer bzw. biochemischer Systeme. Die Betrachtungen in der vorliegenden Arbeit sind zunächst auf Bio-Modelle in der Systembiologie fokussiert, können jedoch auf Bio-Modelle anderer Forschungszweige übertragen werden, wie etwa der Entwicklungsbiologie, der Ökologie, der Populationsbiologie und der Evolutionsbiologie. Zielsysteme von systembiologischen Bio-Modellen sind typischerweise als biochemische Reaktionsnetzwerke oder als Genregulationsnetze konzeptualisiert (siehe unten). Zunehmend spielen auch Multiskalenmodelle in der Systembiologie eine Rolle, in welchen verschiedene Konzeptualisierungsebenen miteinander integriert werden (KITANO 2002a, S. 209).

Eine häufig zu findende Unterteilung der Modellierungsformalismen in der Systembiologie ist entlang der Merkmale deterministisch vs. stochastisch und diskret vs. kontinuierlich (vgl. KLIPP ET AL. 2009, S. 9). Bei deterministischen Formalismen sind durch einen gegebenen Zustand des formalen Systems die Folgezustände eindeutig bestimmt. Bei stochastischen Formalismen entscheiden zufällige Ereignisse über Folgezustände. Diese zufälligen Ereignisse werden entweder explizit betrachtet oder es werden Wahrscheinlichkeitsverteilungen für die möglichen Folgezustände berechnet. Letztere Formalismen heißen auch „probabilistisch“ (DECRAENE ET AL. 2010). Die Unterscheidung diskret/kontinuierlich kann sich entweder auf die möglichen Systemzustände, die Zeit oder auf beides beziehen. Diskrete Formalismen erlauben nur abzählbar viele unterschiedliche Zustände bzw. diskrete Zeitschritte, während bei kontinuierlichen Modellen die Zustandsvariablen bzw. die Zeit reellwertig sind. Es gibt auch hybride Formalismen, in welchen kontinuierliche und diskrete Zustandsänderungen miteinander kombiniert werden (siehe z.B. FISHER ET AL. 2007). ZEIGLER ET AL. 2000 bezeichnet diese als „DEV&DESS“, wobei „DESS“ für Formalismen basierend auf Differenzialgleichungen steht („Differential Equation System Specification“) und „DEVs“ für Formalismen, die auf diskreten Ereignissen basieren („Discrete Event System Specification“). Ein weiteres wichtiges Unterscheidungsmerkmal ist, ob der Raum explizit modelliert wird. Formalismen für die Modellierung räumlicher Vorgänge sind z.B. partielle Differenzialgleichungen und zelluläre Automaten.

In [DECRAENE ET AL. 2010](#) werden existierende Techniken entsprechend fünf fundamentalen Prinzipien der Modellierung und Simulation biochemischer Systeme eingeteilt: (1) deterministische Techniken mit kontinuierlichen Zuständen und Zeit (z.B. gewöhnliche und partielle Differenzialgleichungen), (2) stochastische Techniken (z.B. Markov-Ketten und Chemische Mastergleichung), (3) probabilistische Techniken (z.B. Bayessche Netze und Hidden Markov Modelle), (4) algebraische Techniken mit diskreten Zuständen und Zeit (z.B. Boolesche Netzwerke, Petrinetze, P-Systeme und  $\Pi$ -Kalkulus) und (5) agentenbasierte Techniken (z.B. zelluläre Automaten und stochastische Simulationen). Diese Einteilung ist keine Klassifikation im üblichen Sinne, da die Klassen nicht disjunkt untereinander sind. So gibt es z.B. agentenbasierte, stochastische Modellierungstechniken. Die fünf fundamentalen Prinzipien stellen eher das jeweils charakteristische Merkmal ihrer Vertreter dar. Für Details der angeführten Techniken sei auf die Literatur verwiesen. Einen guten Überblick bieten die einschlägigen Lehrbücher der Systembiologie, wie etwa [KLIPP ET AL. 2009](#). Im Folgenden soll auf weitere wichtige Übersichtsartikel zu systembiologischen Modellierungsansätzen eingegangen werden.

In [DECRAENE ET AL. 2010](#) werden Modellierungsformalismen anhand unterschiedlicher Merkmale miteinander verglichen. Dabei zeigt sich, dass Petrinetze, Prozessalgebren und Bayessche Netze keine explizite Repräsentation von Zeit besitzen. Diese Ansätze sind somit für die Repräsentation zeitlicher Phänomene nur bedingt geeignet. In [MANDEL ET AL. 2004](#) werden verschiedene Ansätze der Modellierung biochemischer Systeme diskutiert und ihre Eignung zur Beschreibung dynamischen Verhaltens untersucht. Der Artikel ist lehrreich, da dasselbe biologische Zielsystem in den verschiedenen Formalismen repräsentiert wird und die jeweiligen Stärken und Schwächen herausgearbeitet werden. Eine sehr detailreiche Klassifikation sowohl stochastischer als auch deterministischer Modellierungsformalismen findet sich in [ULLAH ET AL. 2011](#).

Einen guten Überblick über die unterschiedlichen Ansätze zur diskreten Modellierung biologischer Systeme findet sich in [FISHER ET AL. 2007](#): Es werden Boolesche Netzwerke, Petrinetze, Prozessalgebren und gekoppelte endliche Automaten betrachtet. Da diskrete Modelle direkt in einen Algorithmus übersetzt werden können, der von einem klassischen Rechner verarbeitet werden kann, werden diese Modelle in [FISHER ET AL. 2007](#) als „computational models“ bezeichnet. Im Gegensatz dazu benötigen „mathematische“ (kontinuierliche) Modelle numerische (Näherungs-)Verfahren für die Simulation ihres Verhaltens. „Computational models“ besitzen eine operationale Semantik, während „mathematische“ Modelle eine denotationale Semantik aufweisen ([FISHER ET AL. 2007](#), Box 1, S. 1240). In diesem Sinne wären Differenzengleichungen als eine Mischform anzusehen, da sie zwar eine denotationale Semantik besitzen, jedoch auch direkt algorithmisch gelöst werden können.

Um die Ausführungen über epistemische Konzeptualisierungen und Modellierungsformalismen greifbarer zu machen, werden im Folgenden zwei der bedeutendsten Modellierungsmethoden der Systembiologie im Detail beschrieben: ein deterministisch-kontinuierlicher Ansatz zur kinetischen Modellierung biochemischer Reaktionen und ein algebraischer Ansatz zur Modellierung der Genregulation.

### Kinetische Modelle biochemischer Reaktionsnetzwerke

Das Fachgebiet der Kinetik beschäftigt sich mit der Geschwindigkeit chemischer Reaktionen. Häufig werden auch spezielle Gesetze für die Reaktionsgeschwindigkeit als „Kinetik“

bezeichnet, wie etwa in „Michaelis-Menten-Kinetik“. Kinetische Modelle benutzen solche Geschwindigkeitsgesetze für die Beschreibung zeitlicher Verläufe der Konzentrationen beteiligter Substanzen. Die folgende Darstellung folgt im Wesentlichen [KLIPP ET AL. 2009](#) und [DECRAENE ET AL. 2010](#). Eine kompaktere und abstraktere Darstellung findet sich auch in [IDEKER ET AL. 2001](#) (S. 352).

Biochemische Reaktionsnetzwerke stellen gekoppelte biochemische Reaktionen dar. Dabei repräsentieren die Knoten *chemische Spezies*<sup>1</sup>. Chemische Spezies sind Ansammlungen gleichartiger *molekularen Entitäten*, wie etwa Atome, Moleküle, Ionen und komplexe Verbindungen. Mögliche *chemische Reaktionen* zwischen chemischen Spezies (*Reaktanten*), in welchen andere chemische Spezies (*Produkte*) entstehen, werden durch Hyperkanten von den Reaktant-Knoten zu den Produkt-Knoten dargestellt. Ein biochemisches Reaktionsnetzwerk kann immer auch als Menge *chemischer Reaktionsgleichungen* dargestellt werden.

Zur kinetischen Modellierung biochemischer Reaktionsnetzwerke werden Differentialgleichungen in folgender Weise verwendet: Der Zustand des dynamischen Systems ist durch die Mengen (absolute *Stoffmengen* oder relative *Konzentrationen*) aller chemischen Spezies des Reaktionsnetzwerks zu einem bestimmten Zeitpunkt festgelegt. Für jede chemische Spezies  $S_i$  beschreibt eine Differentialgleichung die zeitliche Änderung der Konzentration  $[S_i]$  (analog für Stoffmenge  $n(S_i)$ ) dieser Spezies in Abhängigkeit vom Systemzustand:

$$\frac{d[S_i]}{dt} = \sum_{j=1}^m \nu_{ij} v_j, \quad (1.1)$$

wobei  $v_j$  die *Reaktionsgeschwindigkeit* der  $j$ -ten Reaktion ist (insgesamt gibt es  $m$  Reaktionen) und  $\nu_{ij}$  die *stöchiometrische Zahl* von  $S_i$  in der  $j$ -ten Reaktion. Für Reaktanten ist die stöchiometrische Zahl negativ, für Produkte positiv. Kommt eine chemische Spezies nicht in einer Reaktion vor, ist die entsprechende stöchiometrische Zahl 0. Mit Hilfe der stöchiometrischen Matrix  $N = \{\nu_{ij}\}$  kann man das Differenzialgleichungssystem kompakt auch wie folgt schreiben:

$$\frac{dS}{dt} = Nv, \quad (1.2)$$

mit dem Zustandsvektor  $S = ([S_1], \dots, [S_n])^T$  des Systems mit  $n$  chemischen Spezies und dem Vektor  $v = (v_1, \dots, v_m)^T$  der Reaktionsgeschwindigkeiten der  $m$  Reaktionen.

Die Geschwindigkeit *elementarer Reaktionen* kann auf Grundlage des Massenwirkungsgesetzes bestimmt werden. Für komplexere Reaktionen, wie etwa enzymatische Reaktionen, werden häufig spezielle Geschwindigkeitsgesetze entwickelt. Sofern der genaue Reaktionsmechanismus bekannt ist, können diese speziellen Geschwindigkeitsgesetze aus dem Massenwirkungsgesetz der beteiligten elementaren Reaktionen abgeleitet werden. Oft wird jedoch auch ein empirisches Geschwindigkeitsgesetz verwendet, dass experimentell ermittelt wird.

Nach dem Massenwirkungsgesetz ist die Reaktionsgeschwindigkeit proportional zur Wahrscheinlichkeit eines Zusammenstoßes der Reaktanten. Die Wahrscheinlichkeit eines Zusammenstoßes ist wiederum proportional zum Produkt der Konzentrationen der Reaktanten, wobei die stöchiometrischen Zahlen als Exponenten der entsprechenden Konzentration

<sup>1</sup>Hervorgehobene chemische Begriffe, die in diesem Abschnitt verwendet werden, können in [IUPAC Gold Book](#) nachgeschlagen werden.

auftreten (KLIPP ET AL. 2009, S. 14). Das entsprechende Geschwindigkeitsgesetz der  $j$ -ten Reaktion lautet:

$$v_j = k_j \prod_{i=1}^n [S_i]^{\nu_{ij}} , \quad (1.3)$$

wobei als Proportionalitätsfaktor der *Geschwindigkeitskoeffizient* (auch als kinetische Konstante bezeichnet)  $k_j$  der  $j$ -ten Reaktion auftritt. Kanten in Reaktionsnetzwerken und Reaktionspfeile in chemischen Reaktionsgleichungen sind meist mit den entsprechenden Geschwindigkeitskoeffizienten beschriftet. Nach Massenwirkungsgesetz schreibt sich das dynamische System wie folgt:

$$\frac{d[S_1]}{dt} = \sum_{j=1}^m \left( \nu_{1j} k_j \prod_{k=1}^n [S_k]^{\nu_{kj}} \right) \quad (1.4)$$

$\vdots$

$$\frac{d[S_n]}{dt} = \sum_{j=1}^m \left( \nu_{nj} k_j \prod_{k=1}^n [S_k]^{\nu_{kj}} \right) . \quad (1.5)$$

Jede Reaktion, an der eine Spezies teilnimmt, taucht als Summand in der Differenzialgleichung auf, die die Konzentrationsänderung der Spezies beschreibt. Dabei hat der Summand ein positives Vorzeichen, wenn die Spezies produziert wird (Produkt in der entsprechenden Reaktion ist) und ein negatives Vorzeichen, wenn die Spezies verbraucht wird (Reaktant in der entsprechenden Reaktion ist).

Handelt es sich bei allen Reaktionen um elementare Reaktionen, ist eine eindeutige Umwandlung eines Reaktionsnetzwerks in ein Differenzialgleichungssystem und umgekehrt mit Hilfe des Massenwirkungsgesetzes möglich. In diesem Fall kann das formale System sowohl durch das Reaktionsnetzwerk als auch durch das Differenzialgleichungssystem spezifiziert werden. Darüber hinaus liegt damit ein Idealfall eines kompetenten Modells vor, da alle elementaren Reaktionen und somit der vollständige Reaktionsmechanismus bekannt sind. Komplexere Geschwindigkeitsgesetze können nicht allein durch ein Reaktionsnetzwerk spezifiziert werden, sondern müssen im Differenzialgleichungssystem angegeben werden. Andererseits enthält ein Differenzialgleichungssystem mit komplexen Geschwindigkeitsgesetzen nicht genug Informationen, um kompetent bezüglich eines Zielsystems zu sein. Im Allgemeinen ergänzen sich erst Reaktionsnetzwerk und Differenzialgleichungssystem zu einem kompetenten Bio-Modell.

Ein Problem bei dieser Vorgehensweise stellen katalytische Reaktionen dar. Ein großer Teil der biochemischen Reaktionen sind durch *Enzyme* katalysiert. Diese enzymatische Katalyse macht die meisten biochemischen Reaktionen in dem Temperaturbereich, in welchem Leben existiert, überhaupt erst möglich bzw. hinreichend schnell. Bei einer katalytischen Reaktion wird der *Katalysator* nicht verbraucht. Oft werden katalytische Reaktionen durch Bruttoreaktionsgleichungen angegeben, in welchen der Katalysator sowohl als Reaktant als auch als Produkt auftaucht. Die entsprechende stöchiometrische Zahl des Katalysators in einer solchen Bruttoreaktionsgleichung ist somit 0. Der Einfluss der Konzentration des Katalysators auf die Reaktionsgeschwindigkeit kann nicht durch das Massenwirkungsgesetz der Bruttoreaktionsgleichung abgebildet werden. Um diesen Einfluss darzustellen, muss der *Reaktionsmechanismus* der enzymatischen Reaktion beachtet werden,



d.h. die Zwischenschritte (elementare Reaktionen) der Bildung des Enzyme-Substrat-Komplexes und der Freisetzung des Produktes aus diesem Komplex (bei enzymatischen Reaktionen wird der Reaktant als „Substrat“ bezeichnet). Dann kann die katalytische Reaktion perfekt durch das Massenwirkungsgesetz beschrieben werden. Meist will man diese Teilreaktionen der enzymatischen Reaktion allerdings nicht betrachten. Dann kann zum Beispiel die *Michaelis-Menten-Gleichung* verwendet werden, um den Einfluss des Katalysators einzubeziehen. Eine Herleitung der Michaelis-Menten-Gleichung und der dafür notwendigen Annahmen findet sich etwa in [KLIPP ET AL. 2009](#) (S. 18f).

Es gibt andere Geschwindigkeitsgesetze, die an die Stelle des Massenwirkungsgesetzes treten können. Eine Verallgemeinerung des Massenwirkungsgesetzes stellt zum Beispiel die „power law kinetics“ dar (vgl. [KLIPP ET AL. 2009](#), S. 29). Abweichungen vom Massenwirkungsgesetz treten auf, wenn Reaktionsschritte unbekannt sind, die Zwischenprodukte experimentell nicht zugänglich sind und andere Faktoren die Reaktionsgeschwindigkeiten beeinflussen. Kinetische Modelle von biochemischen Reaktionsnetzwerken weisen meist eine Mischung verschiedener Geschwindigkeitsgesetze auf. Die Auswahl des richtigen Geschwindigkeitsgesetzes für jede Reaktion stellt einen entscheidenden und schwierigen Schritt der Modellierung dar.

Kurz zusammengefasst kann man für die kinetische Modellierung von biochemischen Reaktionsnetzwerken durch Differenzialgleichungen feststellen:

„Chemical reactions are approximated as continuous deterministic processes at the macroscopic level.“ ([DECRAENE ET AL. 2010](#), S. 1157)

Dieser Approximation liegen verschiedene Annahmen zugrunde: (1) Die festgelegten Geschwindigkeitsgesetze und die kinetischen Konstanten sind zeit-invariant, sie sind also unabhängig vom Systemzustand ([WOLKENHAUER ET AL. 2005](#), S. 1848). (2) Die jeweiligen Anzahlen molekularer Entitäten aller chemischen Spezies ist relativ hoch ([KLIPP ET AL. 2009](#), S. 271f). Bei kleinen Anzahlen molekularer Entitäten hat jedes einzelne Reaktionseignis eine sehr große Bedeutung für das Systemverhalten. Stochastische Simulationen werden dieser Situation besser gerecht. (3) Das Reaktionsmedium ist gut gemischt, alle molekularen Entitäten haben also die gleiche Chance, miteinander zu kollidieren. Ist dies nicht der Fall, spielt die räumliche Anordnung eine wichtige Rolle. Die räumliche Verteilung der Konzentrationen wird in partiellen Differenzialgleichungen explizit repräsentiert, wodurch Diffusionsvorgänge berücksichtigt werden können ([DECRAENE ET AL. 2010](#), S. 1158). Andere Möglichkeiten, räumliche Effekte einzubeziehen, sind die stochastische Simulation der Wege jeder einzelnen molekularen Entität sowie die stochastische Modellierung kleinerer Volumenelemente, welche als gut gemischt angesehen werden, und des Austauschs molekularer Entitäten zwischen diesen Volumenelementen ([KLIPP ET AL. 2009](#), S. 276).

### Boolesche Modelle von Genregulationsnetzen

Kinetische Modelle werden oft eingesetzt, um biochemische Prozesse zu beschreiben, die im Wesentlichen auf der Interaktion von Proteinen beruhen. Das sind insbesondere *metabolische* Systeme. Bei diesen wird ausschließlich Masse transferiert: Proteine werden in enzymatischen Reaktionen erzeugt oder verbraucht ([KLIPP ET AL. 2009](#), S. 92). In biologischen Systemen findet jedoch auch Informationsverarbeitung statt: Genetische

Information wird zur Proteinsynthese benutzt und externe Signale werden verarbeitet (Signaltransduktion).

Die Proteinsynthese auf der Grundlage genetischer Information wird als Genexpression bezeichnet. Die Genexpression erfolgt in mehreren Schritten und wird durch verschiedene Mechanismen reguliert. Eine der wichtigsten Kontrollmechanismen greift in die Transkription ein, also in das Ablesen eines Gens (*DNA*) in Boten-RNA (*mRNA*). Dabei hängt die Genaktivität (in welchem Maße es abgelesen wird) von Transkriptionsfaktoren ab, die an bestimmte Regionen der DNA binden. Diese Transkriptionsfaktoren sind ihrerseits Produkte der Expression anderer Gene. Die Aktivität eines Gens hängt also von der Aktivität anderer Gene ab. Diese Abhängigkeiten werden in Genregulationsnetzen dargestellt: Die Knoten repräsentieren Gene und die gerichteten Kanten regulatorische Einflüsse zwischen den Genaktivitäten. Der Zustand eines Genregulationsnetzes ist durch die Aktivität aller Gene zu einem bestimmten Zeitpunkt definiert. Dies ist eine stark vereinfachte Darstellung. Für eine ausführliche Darstellung siehe [KNÜPFER 2003](#) oder ein beliebiges Genetik-Lehrbuch.

Bei Booleschen Modellen von Genregulationsnetzen hat jedes Gen (Knoten) nur zwei mögliche Zustände: aktiv ('ON', 1), d.h. das Gen wird exprimiert, oder inaktiv ('OFF', 0), d.h. das Gen wird nicht exprimiert. Jedem Knoten mit  $k$  eingehenden Kanten ist eine  $k$ -stellige Boolesche Funktion zugewiesen, die die Aktivität des zugehörigen Gens in Abhängigkeit der Aktivitäten der Gene berechnet, die Quellknoten der eingehenden Kanten sind. Die Berechnung der Zustandsänderungen des Booleschen Netzes erfolgt in diskreten Zeitschritten: Die Aktivitäten aller Gene zum Zeitpunkt  $t + 1$  werden aus den Aktivitäten aller Gene zum Zeitpunkt  $t$  entsprechend der Booleschen Funktionen berechnet. Da hierbei alle Knoten gleichzeitig betrachtet werden, bezeichnet man diese Berechnungsstrategie als „synchron“. Demgegenüber gibt es auch „asynchrone“ Strategien, bei denen in einem Zeitschritt nur der Zustand eines einzelnen Knotens oder einer Gruppe von Knoten berechnet wird. In Booleschen Netzen mit Rückkopplungen (welche häufig in Genregulationsnetzen auftreten), kann sich die Folge von Zuständen zwischen synchroner und asynchroner Berechnung deutlich unterscheiden. Aus diesem Grund gehört, neben den Booleschen Funktionen, die Berechnungsstrategie mit zur Spezifikation des dynamischen Systems. Für Details siehe auch [KNÜPFER 2003](#) und [SMOLEN ET AL. 2000](#) sowie generell zur Modellierung der Genregulation [KLIPP ET AL. 2009](#).

### 1.2.3. Fallbeispiel: Tysons Modelle des Zellzyklus

In diesem Abschnitt sollen zwei dynamische Modelle des Zellzyklus vorgestellt werden, um die obigen Aussagen zu konkretisieren. Die beiden Modelle werden in der gesamten vorliegenden Arbeit als Fallbeispiel dienen und zur Illustration der Wissensfacetten und deren Formalisierung verwendet. Es handelt sich um zwei eng zusammenhängende Modelle aus [TYSON 1991](#): Ein Modell des Zellzyklus mit sechs Variablen (im Folgenden als **Modell 1** bezeichnet) und ein vereinfachtes Modell mit zwei Variablen (**Modell 2**).

Die beiden „TYSON-Modelle“ wurden aus folgenden Gründen als Fallbeispiel ausgewählt:

1. Die TYSON-Modelle sind Modelle des prominenten<sup>2</sup> und bekannten biologischen Systems „Zellzyklus“. TYSON leistete Pionierarbeit in diesem Forschungsgebiet (siehe

---

<sup>2</sup>Im Jahr 2001 gab es einen Medizin-Nobelpreis für die Erforschung der Kontrolle des Zellzyklus für Leland H. Hartwell, R. Timothy Hunt und Paul M. Nurse ([NOBELVERSAMMLUNG 2001](#))



[LE NOVÈRE 2006b](#) für eine kurze Darstellung der Bedeutung der TYSON-Modelle).

2. Die TYSON-Modelle sind überschaubar in der Anzahl der Variablen und Gleichungen. Trotz ihrer scheinbar geringen Komplexität treten erstaunlich viele Probleme bei der Analyse und Formalisierung des involvierten Wissens auf.
3. Es gibt eine Reihe anderer Modelle desselben biologischen Systems, z.B. [GOLDBETER 1991](#). Diese können für semantikbasierten Modellvergleich und Modellkombination mit den TYSON-Modellen benutzt werden.
4. Beide TYSON-Modelle sind in der [BioModels Database](#) ([LE NOVÈRE ET AL. 2006](#)) enthalten. In dieser Datenbank von Bio-Modellen (vgl. [Abschnitt 4.1.1](#)) werden einfache Verlinkungen von Modellkomponenten auf biologische Ressourcen und Ontologien bereitgestellt.
5. Die TYSON-Modelle stehen untereinander in einer Beziehung: **Modell 2** ist eine Vereinfachung von **Modell 1**. Dies ist für die Analyse der Beziehungen zwischen Modellen und deren Formalisierung interessant.

Bevor im Folgenden die beiden TYSON-Modelle und ihre Verwendung dargestellt werden, soll zunächst das biologische System Zellzyklus vorgestellt werden. Ausführliche Darstellungen finden sich in jedem Lehrbuch der Biologie. Bereits im Biologieunterricht in der Schule begegnet man dem Zellzyklus unter den Stichwort Zellteilung (Mitose). Deshalb hier nur in aller Kürze: Der Zellzyklus ist der sich wiederholende Prozess der Zellteilung (Mitose oder M-Phase) und die Periode bis zur nächsten Zellteilung (Interphase). Der Zellzyklus ist ein umfangreicher Prozess, der in verschiedenen Phasen abläuft und einer komplizierten biochemischen Kontrolle und Regulation unterliegt. Einen Überblick über die systembiologische Modellierung des Zellzyklus oder einzelner Teilprozesse findet sich in [KLIPP ET AL. 2009](#).

**Modell 1** ist ein dynamisches Modell für den Zellzyklus. Das Modell beschreibt die Hauptkontrolle des Zellzyklus ohne ins Detail zu gehen. TYSON will zeigen, dass bereits ein so einfaches Modell wichtige Kontrollmechanismen des Zellzyklus erklären kann. Hauptrolle im Modell spielt dabei die Bildung und Aktivierung des **MPF** (*Maturation Promoting Factor*), einem Heterodimer aus den Proteinen Cyclin und **CDK1** (*Cyclin-dependent Kinase 1*). Von [TYSON 1991](#) wird **CDK1** als „cdc2“ bezeichnet, entsprechend dem dieses Protein codierenden Gens. Hier soll die von [UniProt](#) ([APWEILER ET AL. 2004](#)) empfohlene Bezeichnung verwendet werden (siehe [UniProt-Eintrag P04551](#)). Das entsprechende biochemische Reaktionsnetzwerk findet sich in [Abbildung 1.3](#). **Modell 1** modelliert diesen Vorgang mit einem System von sechs Differenzialgleichungen:

$$\frac{d[C2]}{dt} = k_6[M] - k_8[\sim P][C2] + k_9[CP] \quad (1.6)$$

$$\frac{d[CP]}{dt} = -k_3[CP][Y] + k_8[\sim P][C2] - k_9[CP] \quad (1.7)$$

$$\frac{d[pM]}{dt} = k_3[CP][Y] - [pM]F([M]) + k_5[\sim P][M] \quad (1.8)$$

$$\frac{d[M]}{dt} = [pM]F([M]) - k_5[\sim P][M] - k_6[M] \quad (1.9)$$

$$\frac{d[Y]}{dt} = k_1[aa] - k_2[Y] - k_3[CP][Y] \quad (1.10)$$

$$\frac{d[YP]}{dt} = k_6[M] - k_7[YP] . \quad (1.11)$$

Die Konzentrationen der anderen chemischen Spezies in [Abbildung 1.3](#) werden als konstant betrachtet ( $[aa]$  und  $[\sim P]$ , vgl. [TYSON 1991](#), S. 7329) oder spielen in der Untersuchung von [TYSON 1991](#) keine Rolle, da es sich um ein Abbauprodukt handelt ( $P_i$ ). Aus diesem Grund gibt es keine Differenzialgleichungen für die zeitlichen Entwicklungen von  $[aa]$ ,  $[\sim P]$  und  $[P_i]$ .

chemische Spezies	Strukturformel	Bezeichner
<i>Cyclin-dependent Kinase 1</i> (CDK1)	cdc2	<i>C2</i>
phosphoryliertes CDK1	cdc2–P	<i>CP</i>
inaktiver <i>Maturation Promoting Factor</i> (MPF)	P–cdc2–cyclin–P	<i>pM</i>
aktiver <i>Maturation Promoting Factor</i> (MPF)	cdc2–cyclin–P	<i>M</i>
Cyclin	cyclin	<i>Y</i>
phosphoryliertes Cyclin	cyclin–P	<i>YP</i>
Adenosintriphosphat (ATP)	$\sim P$	$\sim P$
anorganische Phosphate	$P_i$	$P_i$
Aminosäuren	aa	<i>aa</i>

Tabelle 1.1.: Chemische Spezies von Modell 1 mit Strukturformeln aus [Abbildung 1.3](#) und Bezeichnern in [Gleichung 1.6](#) bis [Gleichung 1.11](#).

Jede Differenzialgleichung stellt die zeitliche Entwicklung der Konzentrationsänderungen einer chemischen Spezies in Abhängigkeit der Konzentrationen der anderen Spezies dar. Die beteiligte chemische Spezies mit ihren Strukturformeln entsprechend [Abbildung 1.3](#) und ihren Bezeichnern in Modell 1 sind in [Tabelle 1.1](#) zusammengefasst. Außerdem werden die Bezeichner *CT* und *YT* für das gesamte CDK1 bzw. das gesamte Cyclin inklusive des Vorkommens in Dimeren und als phosphorylierte Varianten verwendet:

$$[CT] = [C2] + [CP] + [pM] + [M] \quad (1.12)$$

$$[YT] = [Y] + [YP] + [pM] + [M] . \quad (1.13)$$

Die  $k_i$  stellen kinetische Geschwindigkeitskoeffizienten dar. Die Modellierung der Reaktionen folgt – mit Ausnahme von Reaktion 4 – dem Massenwirkungsgesetz. Reaktion 4 ist die einzige enzymatische Reaktion in Modell 1. Es handelt sich um eine autokatalytische Reaktion, bei der aktiver MPF (cdc2–cyclin–P) seine eigene Bildung katalysiert (dargestellt durch eine gestrichelte Linie in [Abbildung 1.3](#)). TYSON wählt für die Abhängigkeit der Reaktionsgeschwindigkeit dieser autokatalytischen Reaktion von  $[M]$  folgende Gleichung:

$$F([M]) = k'_4 + k_4([M]/[CT])^2 . \quad (1.14)$$

Die Bedeutung von  $k_4$  und  $k'_4$  kann durch die Betrachtung von zwei Extremfällen verdeutlicht werden: Falls kein aktiver MPF vorhanden ist ( $[M] = 0$ ), ist die Reaktionsgeschwindigkeit konstant  $k_4$ . In diesem Fall erfolgt keine Katalyse. Falls andererseits die

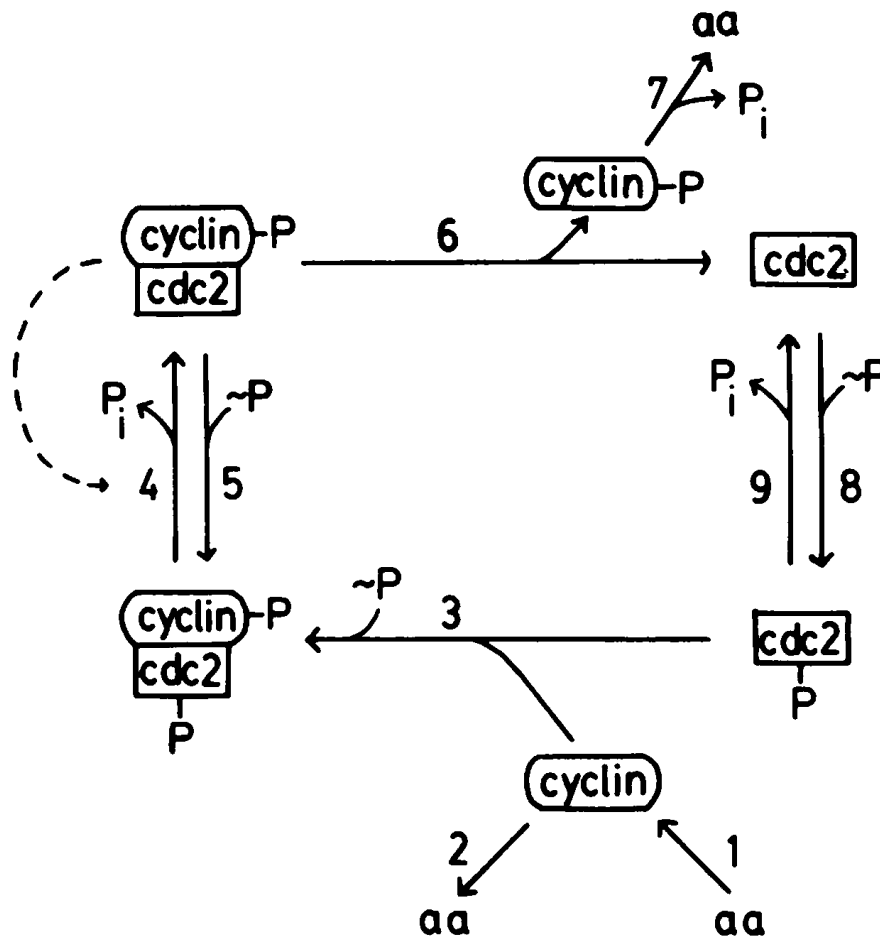


Abbildung 1.3.: Biochemisches Reaktionsnetzwerk von Modell 1 aus [TYSON 1991](#) (Fig. 1, S. 7328). Die Knoten des Netzwerks stellen chemische Spezies dar: Das Protein [CDK1](#) (von [TYSON 1991](#) als „cdc2“ bezeichnet) ist durch ein Rechteck gekennzeichnet, Cyclin durch ein abgerundetes Rechteck. Kombinationen dieser beiden Formen repräsentieren entsprechende Dimere zwischen den Proteinen. Diese können phosphoryliert sein (gekennzeichnet durch „-P“). Weitere chemische Spezies sind unspezifische Aminosäuren („aa“), anorganisches Phosphat („P<sub>i</sub>“) und [ATP](#) („~P“). Eine Zuordnung zwischen der hier verwendeten Darstellung der chemischen Spezies und den in den Gleichungen von Modell 1 verwendeten Bezeichnern findet sich in [Tabelle 1.1](#). Die durchgezogenen gerichteten Hyperkanten repräsentieren Reaktionen mit entsprechender Nummer. Dabei stellen die Startknoten Reaktanten und die Zielknoten Produkte der jeweiligen Reaktion dar. Die gestrichelte Kante repräsentiert einen katalytischen Einfluss auf die Reaktion 4. Für Details siehe Text.

maximale Konzentration an aktivem MPF erreicht ist ( $[M] = [CT]$ ) und zusätzlich die Annahme  $k_4 \gg k'_4$  getroffen wird, ist die Reaktionsgeschwindigkeit gleich  $k'_4$ .  $k_4$  ist also die Geschwindigkeitskonstante der nichtkatalytischen Aktivierung des MPF und  $k'_4$  die Geschwindigkeitskonstante der rein katalytischen Aktivierung des MPF. Die Wahl von Gleichung 1.14 für die Abhängigkeit der Reaktionsgeschwindigkeit von Reaktion 4 von  $[M]$  ist „[...] only one of many possible ways to describe the autocatalytic feedback of active MPF on its own production“ (TYSON 1991, Table 1, S. 7329) und stellt eine kritische Entscheidung bei der Modellierung des Zellzyklus dar, wie in TYSON 1991 (S. 7331) diskutiert wird. Für Details über Modell 1 siehe TYSON 1991.

Modell 2 stellt eine mathematische Vereinfachung von Modell 1 dar, die von TYSON 1991 für die Analyse dynamischer Eigenschaften des Ausgangsmodells herangezogen wird. Nach geeigneten Variablensubstitutionen

$$u = [M]/[CT] \quad (1.15)$$

$$v = ([Y] + [pM] + [M])/[CT] \quad (1.16)$$

und unter bestimmten Annahmen (siehe unten) kann das Gleichungssystem auf zwei Differenzialgleichungen reduziert werden:

$$\frac{du}{dt} = k_4(v - u)(\alpha + u^2) - k_6u \quad (1.17)$$

$$\frac{dv}{dt} = k_1[aa]/[CT] - k_6u \quad (1.18)$$

$$\alpha = k'_4/k_4. \quad (1.19)$$

Dabei vermindert sich auch die Anzahl der freien kinetischen Parameter von zehn ( $k_1 \dots k_9$  und  $k'_4$ ) auf vier ( $k_1$ ,  $k_4$ ,  $k'_4$  und  $k_6$ ). Die genaue Ableitung von Modell 2 aus Modell 1 ist in TYSON 1991 (S. 7331) beschrieben.

Für die Umformung und Vereinfachung von Modell 1 werden folgende Annahmen getroffen (TYSON 1991):

$$k_2 = 0 \quad (1.20)$$

$$k_5 = 0 \quad (1.21)$$

$$k_8[\sim P] \gg k_9 \quad (1.22)$$

$$k_9 \gg k_6 \quad (1.23)$$

$$[CT] = \text{const.} \quad (1.24)$$

Mit den Annahmen gemäß Gleichung 1.20 und Gleichung 1.21 werden die zugehörigen Reaktionen „deaktiviert“, d.h. es wird angenommen, dass Cyclin nicht zerfällt ( $k_2 = 0$ ) und dass aktiver MPF nicht direkt durch Phosphorylierung deaktiviert wird ( $k_5 = 0$ ). TYSON 1991 (S. 7331) diskutiert, dass ein Fallenlassen dieser Annahmen die Ergebnisse nicht grundsätzlich verändern würde. Diese beiden Annahmen erleichtern jedoch die Reduktion von Modell 1 zu Modell 2. Auch die Annahmen gemäß Gleichung 1.22 und Gleichung 1.23 tragen zu dieser Vereinfachung bei, da sie erlauben,  $[C2]$  als konstant anzunehmen und damit Gleichung 1.6 aus Modell 1 zu entfernen: CDK1 wird nach Abspaltung vom aktiven MPF (Reaktion 6) unmittelbar phosphoryliert (Reaktion 8). Die Dephosphorylierung (Reaktion

9) ist wesentlich langsamer als die Reaktion 8, wodurch sich schnell ein Gleichgewicht für  $[C2]$  einstellt (vgl. [TYSON 1991](#), S. 7329). Zusätzlich wird die Annahme getroffen, dass die gesamte Menge **CDK1** gleich bleibt ([Gleichung 1.24](#)). Dies erleichtert die Analyse von Modell 2.

TYSON benutzt Modell 2, um den Einfluss der beiden Parameter  $k_4$  und  $k_6$  auf das Systemverhalten zu untersuchen. Unter Berücksichtigung der Annahmen, stellt das Verhalten von Modell 2 eine Projektion des Verhaltens von Modell 1 dar. Aussagen, die man über das Verhalten von Modell 2 gewinnt, können also auf das Modell 1 übertragen werden. Beide Modelle sind bezüglich der von [TYSON 1991](#) untersuchten Phänomene also gleichermaßen performant (siehe [Abschnitt 1.1.3](#)).

Wie sieht es mit der Kompetenz von Modell 2 aus? Für Modell 2 wird in [TYSON 1991](#) kein biochemisches Reaktionsnetzwerk angegeben. Die Differenzialgleichungen von Modell 2 haben keine sinnvolle Entsprechung in Form eines biochemischen Reaktionsnetzwerks. Die beiden abhängigen Variablen  $u$  und  $v$  repräsentieren nicht direkt die Konzentrationen chemischer Spezies, sondern sind auf  $[CT]$  normalisiert. Problematischer ist es jedoch, dass  $v$  entsprechend [Gleichung 1.16](#) die Konzentrationen mehrerer chemischer Spezies zusammenfasst. Dies verhindert die klare Identifikation chemischer Reaktionen, durch welche  $v$  beeinflusst wird. Zusätzlich beziehen sich sowohl  $u$  gemäß [Gleichung 1.15](#) als auch  $v$  gemäß [Gleichung 1.16](#) auf die Konzentration des aktiven MPF ( $[M]$ ). Modell 2 ist also zwar ein performantes Modell bezüglich der in [TYSON 1991](#) untersuchten Kontrolle des Zellzyklus, aber nicht kompetent bezüglich des biologischen Systems der Bildung und Aktivierung des MPF. Die biologische Interpretation von mit Modell 2 gewonnenen Aussagen kann nur über eine Vermittlung durch Modell 1 erfolgen, entsprechend der Variablensubstitutionen ([Gleichung 1.15](#) und [Gleichung 1.16](#)). Die Vereinfachung eines Modells kann also dazu führen, dass seine Kompetenz vermindert wird oder ganz verschwindet. Das Beispiel von Modell 2 zeigt aber auch, dass mechanistische Erklärungen auch mit Hilfe nicht kompetenter Modelle gewonnen werden können, wenn auch vermittelt über ein kompetentes Modell (in diesem Fall Modell 1).

## Biologische Fragestellungen

Die Fragestellungen, die mit dem Modell untersucht werden sollen, werden in [TYSON 1991](#) explizit benannt:

„Despite such simplifications and omissions, the model in Fig. 1 [vgl. [Figure 1.3](#) – Anmerkung des Autors] is a reasonable ‘first approximation’ to the cell-cycle regulatory network. How good is this picture? Can it account for the coordination of cell growth and division during the normal (somatic) cell cycle? How is the nucleocytoplasmic ratio measured and how is this information communicated to the cyclin-cdc2 mitotic-triggering complex? Can the same model also account for metaphase arrest of unfertilized eggs, for rapid cycles of DNA synthesis and cell division (without cell growth) during the embryonic cell cycle, and for the autonomous cycling of MPF activity in the absence of DNA synthesis or cell division in enucleated embryos?“ ([TYSON 1991](#), S. 7329)

Im Wesentlichen geht es also darum, ob das Modell 1 in der Lage ist, die Kontrolle des Zellzyklus durch MPF widerzuspiegeln. Die Ausübung der Kontrolle des Zellzyklus wird

Größe	Standardwert	Größe	Standardwert
$k_1[aa]/[CT]$	$0.015 \text{ min}^{-1}$	$[C2]$	$0 \text{ mol} \cdot \text{l}^{-1}$
$k_2$	0	$[CP]$	$0.75 \text{ mol} \cdot \text{l}^{-1}$
$k_3[CT]$	$200 \text{ mol}/(1 \cdot \text{min})$	$[Y]$	$0 \text{ mol} \cdot \text{l}^{-1}$
$k_4$	$180 \text{ min}^{-1}$	$[YP]$	$0 \text{ mol} \cdot \text{l}^{-1}$
$k'_4$	$0.018 \text{ min}^{-1}$	$[pM]$	$0.25 \text{ mol} \cdot \text{l}^{-1}$
$k_5[\sim P]$	0	$[M]$	$0 \text{ mol} \cdot \text{l}^{-1}$
$k_6$	$1 \text{ min}^{-1}$	$[CT]$	$1 \text{ mol} \cdot \text{l}^{-1}$
$k_7$	$0.6 \text{ min}^{-1}$	$[YT]$	$0.25 \text{ mol} \cdot \text{l}^{-1}$
$k_8[\sim P]$	$1 \cdot 10^6 \text{ mol}/(1 \cdot \text{min})$	$[u]$	$0 \text{ mol} \cdot \text{l}^{-1}$
$k_9$	$1000 \text{ s}^{-1}$	$[v]$	$0 \text{ mol} \cdot \text{l}^{-1}$
$\alpha$	0.0001		

Tabelle 1.2.: Standardwerte für Parameter (links) und Konzentrationen (rechts) von Modell 1 und Modell 2. Die Werte für die Parameter (außer  $k_8$  und  $k_9$ ) sind [TYSON 1991](#) (Table 2, S. 7329) entnommen. Dabei werden für die als konstant angenommenen Konzentrationen  $[aa]$ ,  $[CT]$  und  $[\sim P]$  die entsprechenden Konstanten um diese Konzentrationen geeignet erweitert. Die Standardkonzentrationen und die Werte für  $k_8$  und  $k_9$  entsprechen den Werten in den Einträgen der [BioModels Database](#) für Modell 1 und Modell 2.

dabei durch Angabe von drei Phänomenen präzisiert, welche das Modell unter bestimmten Bedingungen widerspiegeln soll (Bezeichnung der Phänomene entspricht den Modi in [TYSON 1991](#), siehe unten): A) Hemmung der Metaphase in unbefruchteten Eizellen, B) schnelle Zellteilung ohne Größenwachstum in embryonalen Zellen und C) Koordination von Zellwachstum und Zellteilung in nicht-embryonalen Zellen. Außerdem soll gezeigt werden, dass ein zyklisches Verhalten der Konzentration des aktiven [MPF](#) auch ohne Zellteilung möglich ist. Dieses Verhalten wurde an Eizellen beobachtet, denen der Zellkern entfernt wurde (vgl. [TYSON 1991](#)).

## Simulation und Verhaltensweisen

In diesem Abschnitt wird beschrieben, wie die beiden TYSON-Modelle benutzt werden, um die biologischen Fragestellungen zu adressieren, d.h. konkret die drei oben beschriebenen Effekte der Kontrolle des Zellzyklus widerzugeben. Für die Simulation der TYSON-Modelle werden die Werte der Geschwindigkeitskoeffizienten – bis auf  $k_4$  und  $k_6$  festgelegt ([TYSON 1991](#), Table 2, S. 7329). In [Tabelle 1.2](#) sind die Standardwerte für die Parameter und die Konzentrationen von Modell 1 und Modell 2 angeführt. Im Folgenden werden, sofern nicht anders angegeben, diese Standardwerte verwendet.

Die Parameterwerte sind so gewählt, dass die Annahmen, die für die Ableitung von Modell 2 aus Modell 1 benutzt wurden (siehe oben), erfüllt sind. Das garantiert, dass das zeitliche Verhalten von Modell 2 eine Projektion des zeitlichen Verhaltens von Modell 1 ist, dass also Modell 1 und Modell 2 dieselbe Performanz bezüglich der untersuchten zeitlichen Phänomene besitzen. Dadurch können Aussagen über die Dynamik von Modell 2, die durch Simulation und Analyse gewonnen werden, auf Modell 1 übertragen werden.

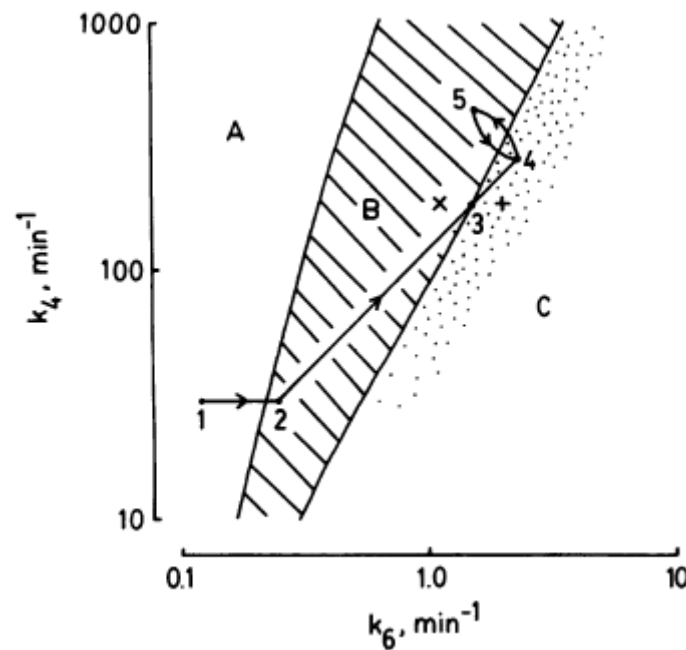


Abbildung 1.4.: Parameterbereiche für unterschiedliche Verhaltensweisen (Modi A, B, C) von Modell 2 in der  $k_4$ - $k_6$ -Ebene aus TYSON 1991 (Fig. 2, S. 7330). Der Pfad 1-2-3-4-5 stellt ein Entwicklungsszenario dar (für Details siehe TYSON 1991, S. 7330): von einer zunächst unbefruchteten Eizelle (1), in der die Metaphase gehemmt ist (Modus A) über die embryonale Phase (2) mit schneller Zellteilung (Modus B), die mit dem sogenannten „midblastula“ Übergang (3) endet, hin zu wachsenden (Modus C) und sich teilenden (Modus B) nicht-embryonalen Zellen (4, 5).

Modell 2 ist einfach genug, dass die Abhängigkeit des Verhaltens von den beiden Parametern  $k_4$  und  $k_6$  analytisch gefunden werden kann. TYSON 1991 (Fig. 2, S. 7330) stellt die drei unterschiedlichen Verhaltensweisen (Modi A, B, C), die das dynamische System zeigen kann, in der  $k_4$ - $k_6$ -Ebene dar (vgl. Abbildung 1.4). Die drei Modi werden zur Beantwortung der biologischen Fragestellungen mit biologischen Verhaltensweisen identifiziert: Modus A (stabiler stationärer Zustand mit hoher Konzentration an MPF) mit der Hemmung der Metaphase in unbefruchteten Eizellen, Modus B (spontane Oszillation) mit der schnellen Zellteilung ohne Größenwachstum in embryonalen Zellen und ein Bereich von Modus C (erregbarer stationärer Zustand) mit der Koordination von Zellwachstum und Zellteilung in nicht-embryonalen Zellen (die Erregbarkeit ermöglicht dabei eine wachstumsbasierte Kontrolle der Zellteilung).

Die Kontrolle des Zellzyklus erfolgt in den TYSON-Modellen durch die Parameter  $k_4$  und  $k_6$ . In diesen Parametern ist der eigentliche Kontrollmechanismus versteckt: Die Reaktionen 4 und 6 werden durch weitere Enzyme („cdc25“ und „wee1“) katalysiert, die nicht in Modell 1 repräsentiert sind. Stattdessen wird deren Einfluss durch die Variation der Geschwindigkeitskoeffizienten  $k_4$  und  $k_6$  abgebildet (TYSON 1991, S. 7331). Dies demonstriert eindrücklich, dass jedes Modell eine Abstraktion ist.



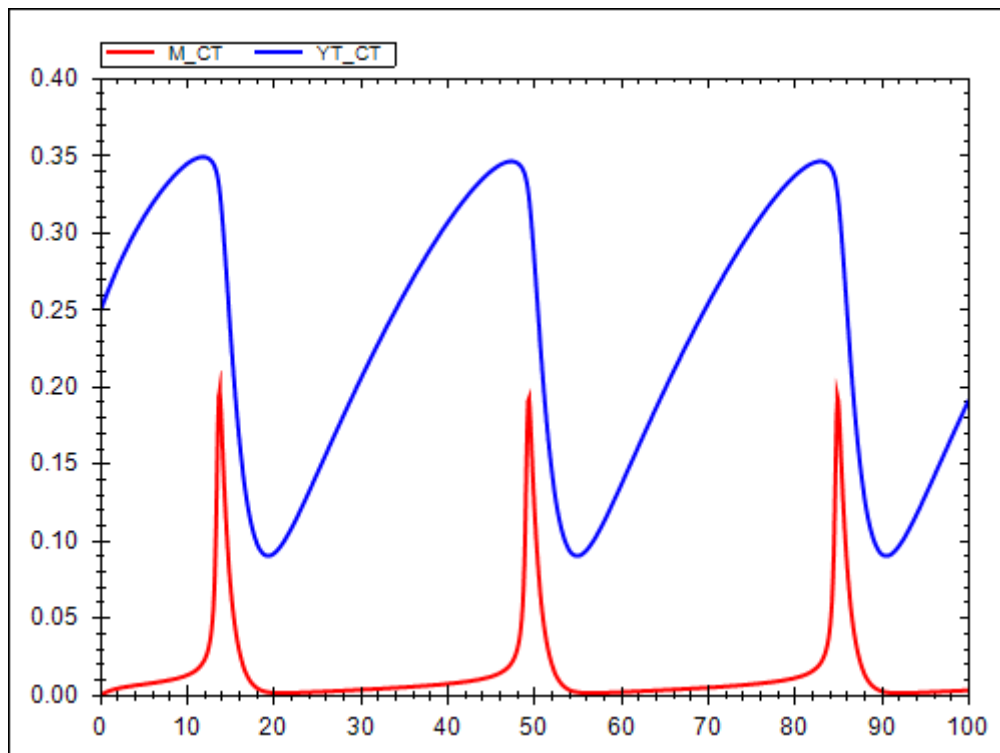


Abbildung 1.5.: Reproduktion der Zeitreihe von Fig. 3a aus TYSON 1991 generiert mittels SED-ML Web Tools ([http://sysbioapps.dyndns.org/SED-ML\\_Web\\_Tools/](http://sysbioapps.dyndns.org/SED-ML_Web_Tools/)). Die Abszissenachse stellt die Zeit (in min) dar, die Ordinatenachse die relativen Konzentrationen  $[M]/[CT]$  (rot) und  $[Y]/[CT]$  (blau). Bei der Simulation wurden die Standardwerte aus Tabelle 1.2 für die Parameter und für die initialen Konzentrationen verwendet.

TYSON 1991 benutzt Modell 1 in Simulationen, um konkretes zeitliches Verhalten des Systems darzustellen. Eine der Simulationen ist in Abbildung 1.5 reproduziert worden. Es zeigt sich ein typisches Verhalten des Modus B (spontane Oszillation).

### Beziehungen zwischen den TYSON-Modellen und zu anderen Modellen

Wie oben beschrieben, kann unter bestimmten Annahmen Modell 1 soweit vereinfacht werden, dass nur noch zwei abhängige Variablen notwendig sind, um das zyklische Verhalten von Zellen zu beschreiben. Modell 2 stellt eine Vereinfachung von Modell 1 dar: Innerhalb bestimmter Randbedingungen weisen beide Modelle dasselbe Verhalten auf. Gleichung 1.15 und Gleichung 1.16 sind nicht direkter Bestandteil von Modell 2, sie stellen aber Beziehungen zwischen den Variablen  $u$  und  $v$  von Modell 2 und den Variablen  $[M]$ ,  $[pM]$  und  $[Y]$  von Modell 1 her.

Nach TYSON 1991 (S. 7332) gibt es verschiedene Beziehungen von Modell 2 zu anderen Modellen: So ähnelt Modell 2 dem Modell von NOREL ET AL. 1991, welches allerdings auf anderen Annahmen basiert. Modell 2 ist identisch mit dem rein hypothetischen oszillatorischen Modell von TYSON ET AL. 1975 und kann als Variante des berühmten „Brüsselators“, einem theoretischen Modell chemischer Oszillation (NICOLIS ET AL. 1977), reformuliert



werden.

Modell 1 ist von diversen existierenden Modellen und aus vorhandenem biologischem Wissen abgeleitet worden (vgl. [TYSON 1991](#)). Auf diese Beziehungen wird hier nicht näher eingegangen.

### 1.3. Computergestützte Modellentwicklung in der Systembiologie

Dieser Abschnitt dient der Präzisierung der Zielvorstellung einer Computerunterstützung der kollaborativen Bildung und Verwendung von Bio-Modellen. Für die Computerunterstützung ist eine formale Repräsentation relevanter Wissensfragmente notwendig, die in den Prozess der Modellierung und Simulation biologischer Systeme involviert sind. Die Zielvorstellung von computergestützter Modellentwicklung in der Systembiologie motiviert somit die folgenden Kapitel dieser Arbeit, die sich mit der Analyse und Formalisierung der für eine Computerunterstützung notwendigen Wissensfragmente beschäftigen.

Die Präzisierung der Vorstellungen einer computergestützten Modellentwicklung in der Systembiologie erfolgt mit Hilfe von Anwendungsszenarien ([Abschnitt 1.3.3](#)). Für ein besseres Verständnis ist es zunächst notwendig, zu untersuchen, wie Bio-Modelle in einem iterativen Prozess gebildet, verwendet und revidiert werden ([Abschnitt 1.3.1](#)). Anschließend werden allgemeine Anforderungen herausgearbeitet, die sich bei der Entwicklung von Bio-Modellen stellen und denen eine Computerunterstützung gerecht werden muss ([Abschnitt 1.3.2](#)). Eine ausführliche Analyse existierender Ansätze zur computergestützten Entwicklung von Bio-Modellen sowie eine Charakterisierung fehlender Formalismen und Verfahren erfolgt in [Kapitel 4](#), nachdem ein geeignetes konzeptuelles Schema für Bio-Modelle vorgestellt wurde (vgl. [Kapitel 2](#)).

Es gibt eine lange Tradition der Auseinandersetzung mit computergestützter Modellentwicklung und -verwendung im Fachgebiet *Modeling and Simulation*. Wichtige Werke sind [KLIR 1985](#), [ZEIGLER 1984](#) und [ZEIGLER ET AL. 2000](#). Explizit mit Computerunterstützung des Modellierungsprozesses setzten sich etwa [SPRIET ET AL. 1982](#) und [CELLIER ET AL. 1982](#) auseinander. [CELLIER ET AL. 1982](#) erweitert das ansonsten typischerweise ingenieurtechnische Anwendungsfeld um Modelle der Biologie, Psychologie sowie Ökonomie. Neuere interessante Ideen der automatischen Modellentwicklung in der Chemie ([SCHAICH ET AL. 2001](#)) und Biologie ([LEIFHEIT ET AL. 2005](#)) scheinen nicht weiter verfolgt zu werden. Einige dieser hier erwähnten traditionellen Ansätze werden in [Abschnitt 2.3](#) detaillierter diskutiert und mit den in dieser Arbeit entwickelten Ideen in Beziehung gesetzt.

Auch im Gebiet der qualitativen Modellierung (siehe [Abschnitt 5.2](#)) wird sich mit der automatischen Generierung von Modellen für physikalisch-technische Systeme auseinandergesetzt, insbesondere in Form der Komposition von Modellfragmenten aus einer Bibliothek zur Lösung konkreter Design- und Diagnoseprobleme (z.B. [FALKENHAINER ET AL. 1991](#) und [IWASAKI ET AL. 1991](#)). Neben dem anderen Anwendungsgebiet dieser Ansätze ist auch der Fokus der qualitativen Modellierung ein anderer als in dieser Arbeit: Die qualitative Modellierung konzentriert sich auf die Wissensrepräsentation physikalischer Systeme, ist also *systemzentriert*. Dagegen untersucht die vorliegende Arbeit die Entwicklung und Verwendung von Modellen, hat also einen *modellzentrierten* Blick. Außerdem verliert die qualitative Modellierung nach ihrer Blütezeit in den 1980ern, wie andere klassische

Zweige der Künstlichen Intelligenz, zunehmend an Bedeutung. Nichtsdestotrotz bietet die qualitative Modellierung einige interessante Ideen zur Repräsentation von Dynamik. Diese werden in [Abschnitt 5.2](#) betrachtet.

### 1.3.1. Modellentwicklungszyklus

In diesem Abschnitt wird beschrieben, wie Modelle eingesetzt werden, um wissenschaftliche Erklärungen zu generieren. Ausgehend von einer Konzeptualisierung des Zielsystems werden Modelle in einem iterativen Prozess der Modellierung, Simulation, Evaluation und Revision immer weiter verbessert. Dieser Prozess startet meist mit bereits bestehenden Modellen oder Modellfragmenten. Sowohl die Konzeptualisierung als auch die Modellevaluation greifen dabei auf experimentelle Beobachtungen des Zielsystems zurück. Damit wird der Modellentwicklungszyklus in den größeren Zusammenhang des wissenschaftlichen Zyklus der hypothetisch-deduktiven Theorieentwicklung eingebettet ([Abbildung 1.6](#)). Im wissenschaftlichen Zyklus werden Modelle selbst zu sich dynamisch ändernden *First-Class*-Objekten, d.h. Modelle werden als Daten behandelt (vgl. [BECKSTEIN ET AL. 2011](#)).

Der wissenschaftliche Zyklus in [Abbildung 1.6](#) beschreibt eine modellbasierte Entwicklung von Theorien (die folgende Darstellung folgt [BECKSTEIN ET AL. 2011](#)). Am Anfang steht dabei die Beobachtung eines Wirklichkeitsausschnitts (Experiment). Die dabei gewonnenen Daten werden auf Regelmäßigkeiten untersucht (Datenanalyse). Auf der Grundlage solcher in den experimentellen Daten gefundener Muster wird ein Modell erstellt. Da die Erstellung eines Modells meist auf bestehende Modelle aufbaut, diese kombiniert, erweitert, verfeinert und verallgemeinert, ist Modellbildung meist auch datenbasierte Modellrevision. Da kompetente Modelle (vgl. [Abschnitt 1.1.3](#)) zur Erklärung natürlicher Phänomene dienen, sind sie wichtige Bausteine wissenschaftlicher Theorien, sie codieren das theoretische Wissen in einer algorithmisierbaren Form. Modelle können sogar selbst als wissenschaftliche Theorie aufgefasst werden:

„A large part of scientific work, and some people believe all scientific work, consists in formalization and model-building.“ ([SPRIET ET AL. 1982](#), S. 1)

Die neu gebildeten bzw. revidierten Modelle tragen in jedem Fall zur Verbesserung der bestehenden Theorien bei, die als Erklärungsrahmen für den untersuchten Wirklichkeitsausschnitt dienen. Für die Verifikation bzw. Falsifikation der verbesserten Theorie müssen nun Hypothesen formuliert werden, die entweder in neuen Experimenten überprüft werden können oder die Grundlage für Simulationen darstellen. Die Simulation eines performanten Modells (siehe [Abschnitt 1.1.3](#)) kann dabei im Sinne von [HEMPEL ET AL. 1977](#) als eine deduktiv-nomologische Form von Erklärung aufgefasst werden, in der mit Hilfe des Modells („Gesetze“ bei [HEMPEL ET AL. 1977](#)) aus den Hypothesen („Bedingungen“ bei [HEMPEL ET AL. 1977](#)) auf zeitliches Verhalten des formalen Systems („Explanandum“ bei [HEMPEL ET AL. 1977](#)) geschlossen wird. Die Ergebnisse der Simulationen gestatten also Prädiktionen über das Verhalten des Zielsystems. Ein Vergleich zwischen den Daten, die einerseits aus dem Experiment und andererseits aus der Simulation stammen, bildet die Grundlage für weitere Modellrevisionen. Dabei werden in der Regel nicht die Rohdaten direkt verglichen, sondern schon angemessen aufbereitete Daten bzw. Muster aus der Datenanalyse. Bezogen auf die Systembiologie stellt sich dieses Wechselspiel zwischen Datenerhebung und -analyse

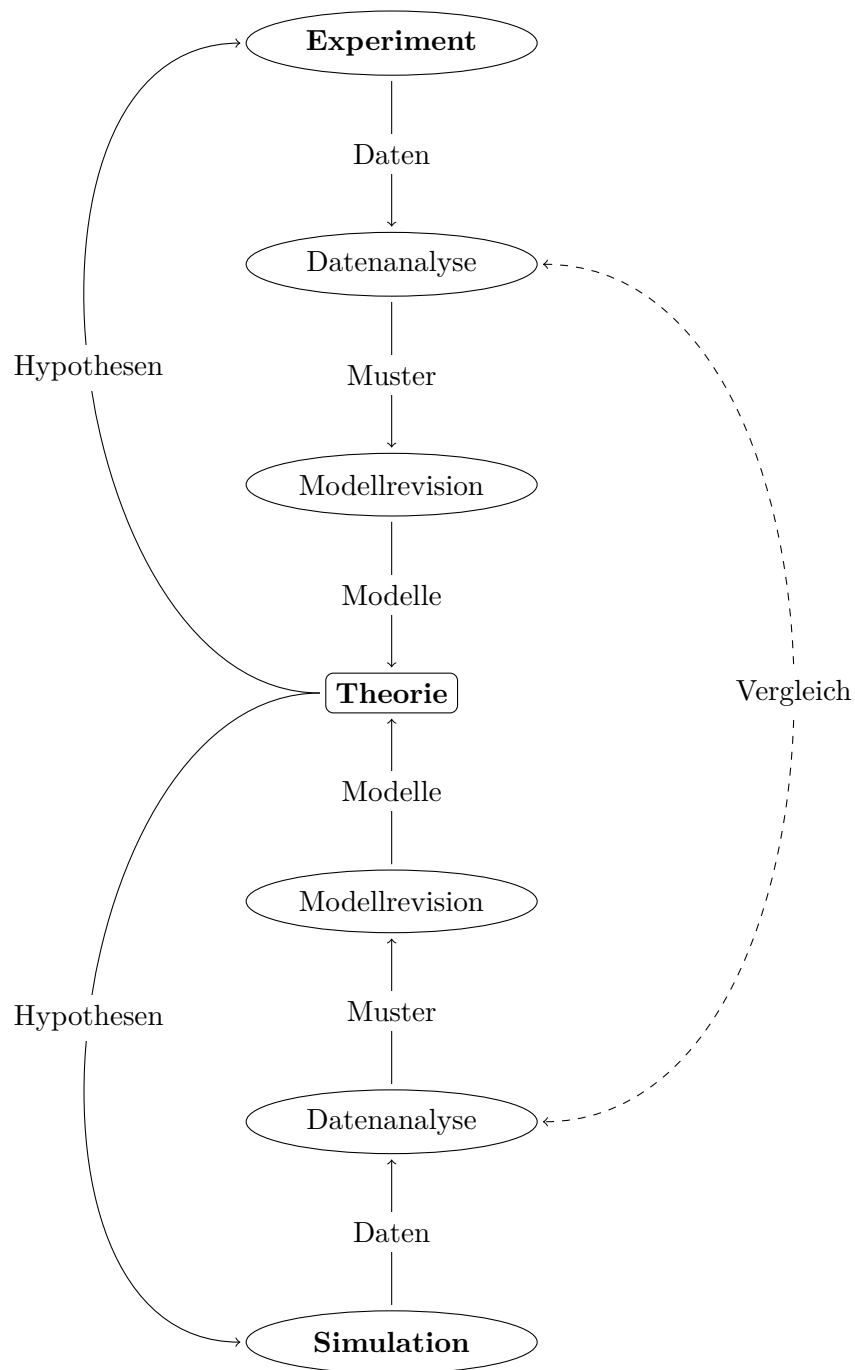


Abbildung 1.6.: Wissenschaftlicher Zyklus nach [BECKSTEIN ET AL. 2011](#) (Abbildung 1, S. 2).

(„discovery-driven“) und Modellrevision und Hypothesengenerierung („hypothesis-driven“) wie folgt dar:

„Moreover, systems biology involves an iterative, strategic interplay between discovery- and hypothesis-driven science. Global observations (discoveries) are matched against model predictions (hypotheses) in an iterative manner, leading to the formation of new models, new predictions, and new experiments to test them.“ (IDEKER ET AL. 2001, S. 355)

Eine explizite Referenz auf die hypothetisch-deduktive Theorieentwicklung in der Systembiologie findet sich als „cycle of knowledge“ in KELL ET AL. 2004 (Figure 1, S. 100) als iterativer Prozess zwischen Wissen und Beobachtungen.

Abbildung 1.6 weist eine gewisse Symmetrie zwischen dem unteren und dem oberen Teilzyklus auf. Aus Sicht der Datenanalyse und Modellrevision ist es mehr oder weniger unwichtig, woher die Daten stammen (BECKSTEIN ET AL. 2011). Dies verwundert insofern nicht, da ja einer der Hauptgründe für die Verwendung dynamischer Modelle (vgl. Abschnitt 1.1.2) darin liegt, das Experimentieren am natürlichen System durch Simulation eines formalen Systems zu ersetzen.

### Schritte der Modellentwicklung

Wird die Modellentwicklung als eigenständiger Prozess betrachtet, lassen sich folgende Teilschritte identifizieren:

- (0) Die *Konzeptualisierung* des Wirklichkeitsausschnitts als Zielsystem schließt die Auswahl relevanter Merkmale und Beziehungen des Zielsystems ein.
- (1) Die *Modellbildung* bzw. *Modellrevision* stellt den konzeptualisierten Wirklichkeitsausschnitt als formales System dar bzw. ändert diese Darstellung. Dies beinhaltet die Erstellung eines Netzwerks und die Wahl geeigneter Mechanismen für die dynamischen Beziehungen im Netzwerk.
- (2) Die *Modellimplementierung* erstellt die entsprechende Spezifikation des formalen Systems in einer passenden computerverarbeitbaren Sprache.
- (3) Die *Modellverwendung* besteht in der Analyse des Modells bzw. in der Simulation des Modells unter geeigneten Rand- und Anfangsbedingungen. Dies kann auch eine Parameteroptimierung bezüglich experimenteller Daten einschließen. An die Simulation schließt sich die Aufarbeitung der Simulationsdaten, ihre Analyse und Interpretation an.
- (4) Die *Modellevaluation* bewertet die Ergebnisse der Simulation. Die Bewertung der Performanz erfolgt durch einen Vergleich mit den entsprechend analysierten experimentellen Daten. Neben der Performanz können noch andere Qualitätskriterien zur Bewertung eines Modells herangezogen werden, wie etwa Komplexität, Stabilität und Sensitivität des Modells.
- (5) Ist die Modellevaluation noch nicht zufriedenstellend, beginnt eine neue Iteration mit Punkt 1. Dafür ist es hilfreich, die Ursachen für die Mängel des Modells zu identifizieren, um eine zielgerichtete Modellrevision durchführen zu können.

Das hier beschriebene Vorgehen der Modellentwicklung ist typisch für die *Bottom-up* Systembiologie (vgl. [Abschnitt 1.2](#)). Jedoch werden auch in der *Top-down* Systembiologie diese Schritte durchlaufen, wenn auch mit unterschiedlichen Zielsetzungen und verschiedenen Methoden der Modellbildung ([Punkt 1](#)): Ein zunächst durch statistische Verfahren aus den Daten generiertes performantes Modell wird schrittweise zu einem kompetenten Modell verfeinert. Dabei müssen entsprechende Kriterien der Kompetenz in die Modellevaluation einfließen.

Die Modellevaluation kann neue Experimente am Zielsystem notwendig machen, etwa wenn die vorhandenen Beobachtungsdaten nicht ausreichen, einen Vergleich verschiedener Modellkandidaten bezüglich ihrer Performanz vorzunehmen. Die Durchführung neuer Experimente führt zu einer neuen Iteration des umfassenden wissenschaftlichen Zyklus (siehe oben).

Die Konzeptualisierung des Wirklichkeitsausschnitts steht vor der eigentlichen Modellentwicklung. Es kann jedoch sein, dass eine Modellrevision auch eine Anpassung der Konzeptualisierung erfordert, z.B. durch die Auswahl anderer relevanter Merkmale. Aus diesem Grund wurde die Konzeptualisierung hier mit angeführt. Der weitverbreitete Begriff der *Modellierung* umfasst in der hier verwendeten Terminologie die Schritte Konzeptualisierung ([Punkt 0](#)), Modellbildung ([Punkt 1](#)) und Modellimplementierung ([Punkt 2](#)). Wie oben angeführt, wird in dieser Arbeit davon ausgegangen, dass dynamische Modelle in Simulationen verwendet werden. Damit kann der Modellentwicklungszyklus kurz zusammengefasst werden als: Modellierung – Simulation – Evaluation.

Eine Darstellung der Modellentwicklung als iterativer Prozess findet sich in ähnlicher Form in zahlreichen Publikationen zu dynamischen Modellen. Eine grobe Darstellung als „systems biology cycle“ findet sich in [DOYLE ET AL. 2006](#) (S. 603 und Figure 1, S. 604) als Wechselspiel zwischen Experimenten, Modellierung und Hypothesengenerierung. In [CHELLIAH ET AL. 2009](#) (Fig. 2, S. 11) wird in Anlehnung an die Softwareentwicklung der „life cycle of systems biology models“ präsentiert. Hier werden nicht die Teilprozesse angeführt, sondern die jeweiligen Ergebnisse: Der Zyklus geht vom biochemischen Reaktionsnetzwerk zum mathematischen Modell, weiter zum implementierten Modell (in [CHELLIAH ET AL. 2009](#) als „computational model“ bezeichnet) und schließlich zur Simulation und beginnt dort von neuem. Eine der detailliertesten Darstellungen findet sich in [SPRIET ET AL. 1982](#) (Fig. 1.2, S. 5), vgl. [Abbildung 1.7](#). Die einzelnen Prozesse in dieser Abbildung können den oben angegebenen Teilschritten der Modellentwicklung zugeordnet werden.

#### 1.3.2. Allgemeine Anforderungen für die Entwicklung von Bio-Modellen

Die Systembiologie stellt aufgrund ihrer untersuchten biologischen Systeme einige besondere Anforderungen an die Modellierung und Simulation:

1. Biologische Systeme sind hierarchisch organisiert. Ein umfassendes Verständnis solcher Systeme verlangt auch *Multiskalenmodelle* (vgl. das [Diktum](#) von [IDEKER ET AL. 2001](#) am Anfang dieses Kapitels). Die Modellbildung eines Multiskalenmodells vollzieht sich meist durch Integration einzelner Modelle für die verschiedenen Ebenen. Dafür müssen entsprechende Schnittstellen zwischen den Teilmodellen definiert werden, was meist auch eine Anpassung der Teilmodelle an diese besondere Anforderung mit sich bringt. Außerdem sind eventuell Prozeduren erforderlich, die Daten der einzelnen

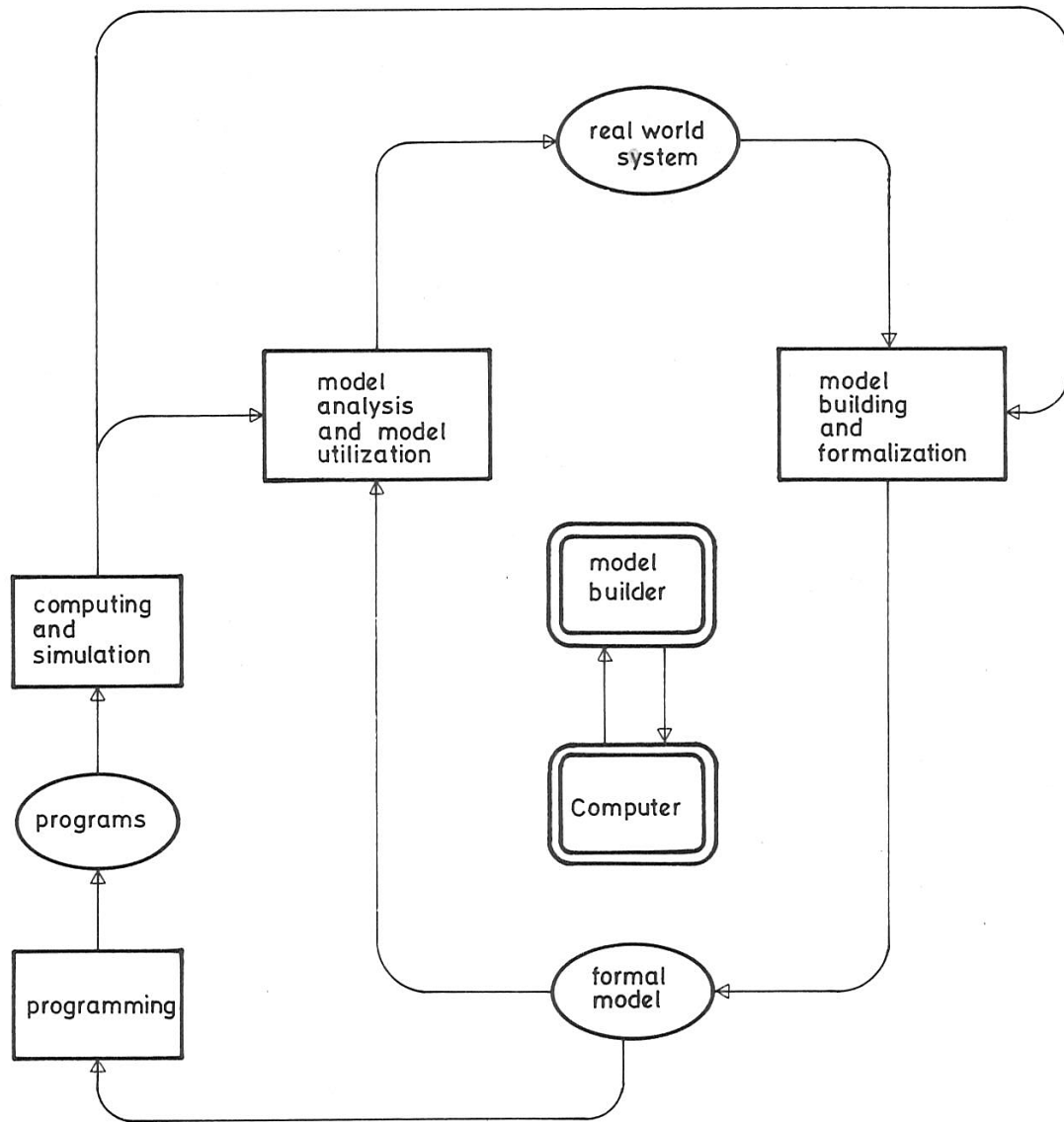


Abbildung 1.7.: Modellbildung und Modellverwendung, aus [SPRIET ET AL. 1982](#) (Fig. 1.2, S. 5). Ellipsen stellen Gegenstände dar, Rechtecke Tätigkeiten. Die Kanten verbinden die Gegenstände mit entsprechenden Tätigkeiten, in denen sie benutzt bzw. erzeugt werden. Durchgeführt werden die Tätigkeiten von einem menschlichen Modellierer („model builder“) mit Hilfe eines Computers (beide Akteure und deren Interaktion sind in der Mitte dargestellt).

Teilmodelle geeignet aggregieren. Die Teilmodelle für die unterschiedlichen Ebenen basieren darüber hinaus in der Regel auf unterschiedlichen Formalismen. Die resultierenden hybriden Multiskalenmodelle stellen eine besondere Herausforderung für die Simulation dar. Auch die Identifizierung von Ursachen für eine nicht ausreichende Performanz ist bei Multiskalenmodellen erheblich schwieriger und damit auch eine zielgerichtete Modellrevision zur Verbesserung der Performanz.

2. Das Ziel von Simulationen in der Systembiologie sind oftmals nicht die genauen numerischen Werte des zeitlichen Verhaltens, sondern *qualitative Beschreibungen* des Modellverhaltens, etwa durch eine Klassifikation von Verhaltensweisen:

„Nowadays, many computational biologists avoid modeling-as-data-fitting, opting instead to create models in which networks are specified in terms of elements and interactions (the network ‚topology‘), but the numerical values that quantify those interactions (the parameters) are deliberately varied over wide ranges. As a result, the study of such networks focuses not on the exact values of outputs, but rather on qualitative behavior, e.g., whether the network acts as a ‚switch‘, ‚filter‘, ‚oscillator‘, ‚dynamic range adjuster‘, ‚producer of stripes‘, etc.“ (LANDER 2004, S. 0712)

Dies liegt in der Tradition der Theorie dynamischer Systeme (vgl. [Abschnitt 5.1](#)), die durch ein Streben nach abstrakten qualitativen Charakterisierungen von Systemverhalten gekennzeichnet werden kann. Die Modellverwendung muss also entsprechende qualitative Beschreibungen des Modellverhaltens generieren.

3. Die Entwicklung von Bio-Modellen geschieht in der Regel auf der Grundlage bereits existierender Modelle, d.h. durch Modifikation bzw. *Kombination* existierender Modelle oder Modellfragmente. Die Modellkombination kann durch Koppelung von Teilmodellen erfolgen oder durch Integration (Einbettung) eines Modells in ein anderes. Diese Art der Modellentwicklung basiert also auf der Wiederverwendung von Modellen, wodurch besondere Anforderungen an die Speicherung, Spezifikation und Dokumentation von Modellen gestellt werden.
4. Die enge Interaktion zwischen biologischen Experimenten und computergestützten Analyse- und Simulationsverfahren, die kennzeichnend für die Systembiologie ist, macht die Modellentwicklung von vorn herein zu einem *kollaborativen* Unterfangen zwischen experimentellen und theoretischen Biologen einerseits und Mathematikern und Informatikern andererseits:

„The construction and testing of quantitative representations and models will be possible through the collaborative input of experimental and theoretical biologists working together with system analysts, computer scientists, mathematicians, engineers, physicists, and physicians to contend creatively with the hierarchical and nonlinear nature of cellular systems [...]“ (KRIETE ET AL. 2006, S. ix)

Kollaboration findet auch in den einzelnen Phasen der Modellentwicklung statt. Bio-Modelle werden heute oft von global verteilten Gruppen gemeinsam entwickelt. Die Einbeziehung verteilt agierender Wissenschaftler, teilweise aus weit entfernten



Fachdisziplinen, erzeugt besondere Herausforderungen an den Austausch, die Verwendbarkeit und die Dokumentation von Modellen sowie an die Protokollierung des Modellentwicklungsprozesses (Stichwort Versionsverwaltung).

Bevor im nächsten [Abschnitt 1.3.3](#) Teilaufgaben der computergestützten Modellentwicklung betrachtet werden, die einzelne Teilschritte des Entwicklungsprozesses betreffen, soll im Folgenden auf Funktionalitäten einer computergestützten Modellentwicklung eingegangen werden, die im gesamten Entwicklungsprozess eine Rolle spielen bzw. den gesamten Prozess selbst betreffen:

**Entscheidungsunterstützung** Angefangen von der Wahl der richtigen epistemischen Ebene der Konzeptualisierung des Zielsystems, über die Selektion relevanter Merkmale, der Auswahl eines geeigneten Modellierungsformalismus und von Simulationsverfahren, bis zur Festlegung geeigneter Simulationen unter plausibler Rand- und Anfangsbedingungen sowie der Analyse der Resultate – der Prozess der Modellentwicklung ist voller Entscheidungen. Eine Computerunterstützung bei diesen Entscheidungen (*Decision Support*) wird umso wichtiger, je mehr sich die verfügbaren Verfahren, Beschreibungssprachen, Algorithmen und Programme vermehren und die Modelle in Größe und Komplexität zunehmen.

**Integritätssicherung** Während der gesamten Modellentwicklung sind bestimmte Integritätsbedingungen zu sichern. Dies betrifft sowohl syntaktische (Korrektheit) als auch semantische Aspekte (Gültigkeit). Mit Hilfe des hier entwickelten konzeptuellen Schemas der Wissensfacetten ([Kapitel 2](#)) ist es möglich, verschiedene Korrektheits- und Gültigkeitskriterien für Bio-Modelle zu definieren.

**Protokollierung** Die Schritte der Modellentwicklung und die dabei getroffenen Entscheidungen sollten protokolliert werden, d.h. es sollte aufgezeichnet werden, was wann gemacht wurde und warum. Dadurch wird der Prozess der Modellierung und Simulation nachvollziehbar. Änderungen an Modellspezifikationen und Simulationsbeschreibungen werden idealerweise durch eine Versionsverwaltung verfolgt, insbesondere in einem kollaborativen Szenario der Modellentwicklung.

**Dokumentation** Die im Entwicklungsprozess generierten Beschreibungen und Daten sollten gut dokumentiert sein. Die Dokumentation umfasst zum einen Metainformationen über die Entstehung der Beschreibungen und Daten, also etwa Autorenschaft, Datum, verwendete Werkzeuge und Verweise auf entsprechende Protokolle. Zum anderen muss die Dokumentation klar festlegen, wie die Beschreibungen und Daten interpretiert und verwendet werden können.

**Annotation** Eine bestimmte Form der Dokumentation stellen Annotationen dar: Annotationen sind maschineninterpretierbare Informationen über Komponenten von Beschreibungen und Daten (siehe auch [Abschnitt 3.1.2](#)). Computerunterstützung bei der Erstellung von Annotationen sichert deren Vollständigkeit und Korrektheit.

Die Entscheidungsunterstützung führt zu einem geleiteten Vorgehen. Durch die Integritätssicherung wird der Entwicklungsprozess weniger fehleranfällig, die generierten Beschreibungen und Daten sind syntaktisch korrekt und semantisch konsistent. Insbesondere wird somit gesichert, dass die entwickelten Modelle kompetent bezüglich des Zielsystems



sind. Durch die Protokollierung, Dokumentation und Annotation stehen im Ergebnis eines solchen computergestützten Entwicklungsprozess wiederverwendbare, verständliche Modellspezifikationen, reproduzierbare Simulationen sowie verwendbare und bewertbare Simulationsresultate. Zusätzlich wird der gesamte Entwicklungsprozess nachvollziehbar und die getroffenen Entscheidungen transparent.

#### 1.3.3. Teilaufgaben einer computergestützten Modellentwicklung

In diesem Abschnitt sollen einige Teilaufgaben bezüglich der Entwicklung von Bio-Modellen vorgestellt werden, bei denen eine Computerunterstützung erstrebenswert erscheint. Diese Darstellung ist eine Erweiterung von [KNÜPFER ET AL. 2013b](#). Elementare Aufgaben für die computergestützte Modellentwicklung sind:

**Modellsuche** Auf der Grundlage gegebener Eigenschaften sollen Modelle gefunden werden, die diese Eigenschaften besitzen. Dabei soll nicht nur die Struktur des Modells einbezogen werden, sondern auch seine vorgesehene Verwendung und sein Verhalten.

**Modellvergleich** Der Vergleich zweier gegebener Modelle zielt etwa auf die Beantwortung folgender Fragen: Überlappen die beiden Modelle semantisch, d.h. bezüglich ihrer biologischen Interpretation? Ist ein Modell ein Teil-Modell des anderen? Ist ein Modell eine Vereinfachung des anderen? Neben dem Vergleich der Modellstruktur können Modelle auch bezüglich ihrer Verwendung (Funktion) und ihres Verhaltens miteinander verglichen werden. Methoden für den Modellvergleich sind notwendig für die Ausführung von Aufgaben auf höheren Ebenen, z.B. Modellintegration und Modellbewertung.

**Modellbildung/Modellmodifikation** Die Bildung eines neuen Modells oder die Modifikation eines existierenden Modells soll auf einer möglichst hohen Beschreibungsebene möglich sein, ohne direkt auf den Code einer konkreten Implementierung des Modells zugreifen zu müssen. Modifikationen werden dabei durch Annotationen von Modellkomponenten bezüglich ihrer biologischen Bedeutung wesentlich vereinfacht.

**Modellverwendung** Jeder Verwendung eines Modells liegt eine bestimmte biologische Fragestellung zugrunde. Ausgehend von dieser Fragestellung soll die Festlegung einer passenden Modellverwendung und die Durchführung der entsprechenden Simulation unterstützt werden.

**Modellbewertung** Die Analyse der Simulationsresultate und eine entsprechende Bewertung des Modells anhand vorgegebener Qualitätskriterien soll, soweit möglich, automatisch erfolgen. Dies setzt eine Objektivierung der Qualitätskriterien voraus.

Die folgenden höheren Aufgaben der Modellentwicklung können auf die elementaren Aufgaben zurückgeführt werden:

**Modellkombination** Die Kopplung bzw. Integration mehrerer Modelle verlangt zunächst die Angabe der Schnittstellen. Mögliche Kandidaten für solche Schnittstellen können aufgrund semantischer Überlappungen der Teilmodelle gefunden werden, z.B. sind dies Modellkomponenten, die dasselbe biologische Objekt repräsentieren oder

biologische Objekte, die in einer biologischen Hierarchiebeziehung stehen. Bei dem Zusammenfügen der Teilmodelle muss die Integrität des Gesamtmodells gesichert werden.

**Fehlerdiagnose** Ist die Evaluation des verwendeten Modells bezüglich untersuchter biologischer Phänomene und anderer Qualitätskriterien nicht zufriedenstellend, müssen die Ursachen für diesen Fehler gesucht werden. Eine entsprechende Fehlerdiagnose liefert Hinweise, wie das Modell bzw. seine Verwendung verbessert werden können.

**Modellevolution** Auf der Basis ausreichend quantifizierter Qualitätskriterien ist eine „Evolution“ von Modellen mit entsprechenden Optimierungsverfahren vorstellbar. Ein solches Vorgehen liefert vielleicht nicht ein fertiges Modell für eine biologische Fragestellung, kann jedoch gute Modellkandidaten bereitstellen, die als Ausgangspunkt für die weitere Modellentwicklung dienen können.

**Modellzugänglichkeit** Eine Präsentation eines Modells, seiner Verwendung und des dabei gezeigten Verhaltens auf einer hohen Beschreibungsebene, die grafische Elemente verwendet und biologisches Wissen einbezieht, erhöht wesentlich die Zugänglichkeit zu einem Modell. Insbesondere können die grundlegende Idee und Motivation des Modells so auch jenen vermittelt werden, die nicht mit mathematischen Formalismen vertraut sind.

### Beispiel 1.1 (Modellsuche und Modellmodifikation)

Die Modellsuche soll verschiedene Aspekte dynamischer Modelle (vgl. [Kapitel 2](#)) einbeziehen. So sollten beispielsweise beide TYSON-Modelle bei folgenden Anfragen gefunden werden:

- „Finde Modelle, die den Zellzyklus beschreiben!“ Diese Anfrage benötigt Wissen über das modellierte Zielsystem.
- „Finde Modelle mit Bezug zur Proteinkinase *p34*!“ Diese Anfrage erfordert Wissen über die biologische Bedeutung der Modellkomponenten, über Synonyme dieser Komponenten (*p34* ist ein Synonym für das Protein [CDK1](#), das durch *C2* in [Modell 1](#) repräsentiert ist, vgl. [TYSON 1991](#), S. 7328) sowie über die Beziehungen zwischen den beiden Modellen.
- „Finde Modelle, die die Kontrolle des Zellzyklus durch *wee1* untersuchen!“ Diese Anfrage benötigt Wissen über die intendierte Verwendung der Modelle: Die Untersuchung der Abhängigkeit des Verhaltens der TYSON-Modelle vom Parameter  $k_6$ , welche einen Einfluss des Enzyms *wee1* auf den Zellzyklus verkörpert (vgl. [TYSON 1991](#), S. 7331).
- „Finde Modelle, die sowohl einen stationären Zustand als auch ein zyklisches Verhalten aufweisen!“ Diese Anfrage erfordert Wissen über mögliche Verhaltensweisen der Modelle.

Wie kann eine computergestützte Modifikation von [Modell 1](#) ablaufen? Angenommen, [Modell 1](#) soll so modifiziert werden, dass der Einfluss des Enzyms *wee1* auf Reaktion 6 genauer dargestellt wird. [Modell 1](#) könnte wie folgt computergestützt modifiziert werden:

(1) Die Reaktion 6 wird zu einer katalytischen Reaktion geändert. (2) Die Entwicklungsumgebung fragt nach dem entsprechenden Enzym. Die Angabe von *wee1* erzeugt einen neuen Knoten im Reaktionsnetzwerk und eine neue Differenzialgleichung, die die zeitliche Konzentrationsänderung für *wee1* repräsentiert. Entsprechende Annotationen für *wee1* werden erstellt. (3) Das kinetische Gesetz für Reaktion 6 wird abgefragt. Hier kann aus vorgegebenen Gesetzen gewählt oder eine eigene Gleichung angegeben werden. Die Differenzialgleichungen für  $[M]$ ,  $[YP]$  und  $[C2]$  werden entsprechend angepasst. (4) Die zeitliche Konzentrationsänderung für *wee1* muss angegeben werden. Hierfür werden neue Reaktionen ins Reaktionsnetzwerk aufgenommen, die *wee1* bilden oder verbrauchen. Dabei sind eventuell weitere chemische Spezies zuzufügen, was wiederum weitere Änderungen des Reaktionsnetzwerks und des Differenzialgleichungssystems nach sich zieht, die in analoger Weise durchgeführt werden, bis das Modell vollständig ist. ■

## Zusammenfassung

In diesem Kapitel wurde der Begriff des Bio-Modells eingeführt ([Abschnitt 1.2](#)). Bio-Modelle werden in der Systembiologie insbesondere dazu verwendet, dynamisches Verhalten biologischer Systeme zu untersuchen und mechanistisch zu erklären. Bei der dynamischen Modellierung ([Abschnitt 1.1](#)) wird ein Zielsystem durch ein formales System codiert. Dynamische Modelle dienen dabei der Spezifikation entsprechender formaler Systeme. Simulationen des formalen Systems können in Aussagen über das Verhalten des Zielsystems decodiert werden. Damit ist das formale System performant bezüglich der modellierten zeitlichen Phänomene. Das dynamische Modell muss zusätzlich kompetent sein, um als mechanistische Erklärung der zeitlichen Phänomene dienen zu können, d.h es muss eine strukturelle Ähnlichkeit zwischen formalem System und Zielsystem bestehen. Abschließend wurden die Abläufe bei der Modellierung, Simulation und Evaluation von Bio-Modellen beschrieben und Anforderungen an eine computergestützte Modellentwicklung in der Systembiologie formuliert ([Abschnitt 1.3](#)). Mit Hilfe von [Abbildung 1.8](#) sollen noch einmal die wichtigsten Begriffe dieses Kapitels zusammengefasst werden. Dabei lassen sich drei Betrachtungsweisen von Bio-Modellen identifizieren: Struktur, Funktion und Verhalten. Diese sind die tragenden Säulen des im folgenden [Kapitel 2](#) entwickelten konzeptuellen Schemas der Wissensfacetten von Bio-Modellen.

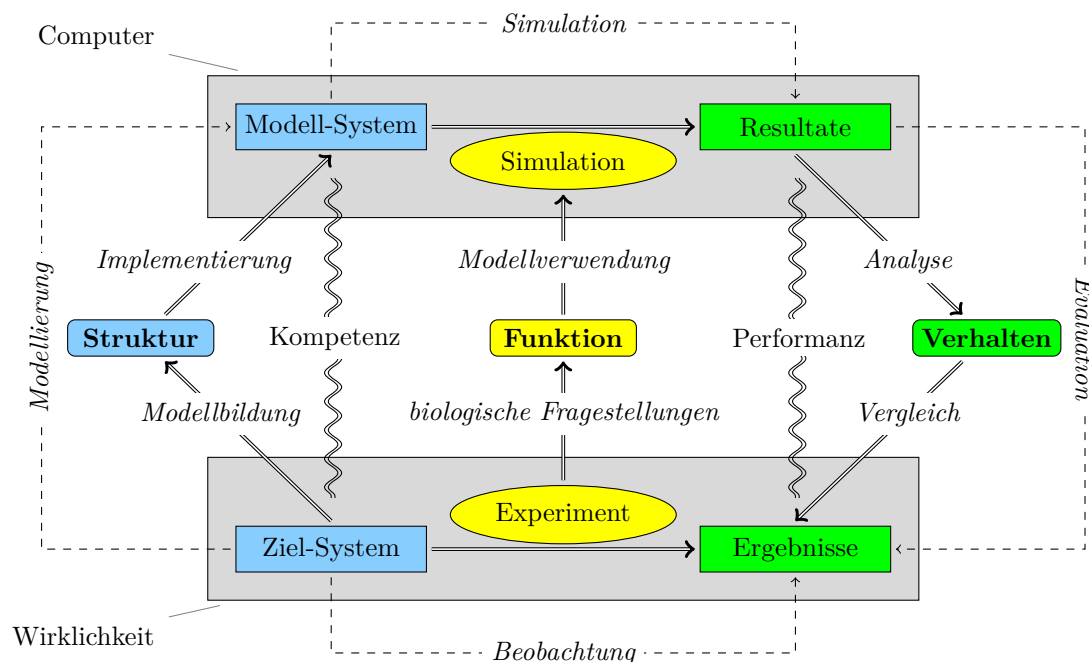


Abbildung 1.8.: Wichtige Konzepte der dynamischen Modellierung und Simulation. Ein formales Modell (verkörpert durch die mittlere Ebene) vermittelt durch eine *Modeling Relation* (vgl. [Abschnitt 1.1.1](#)) zwischen einem Zielsystem (untere Ebene, „Wirklichkeit“) und einem formalen System (obere Ebene, „Computer“). Bei dynamischen Modellen lassen sich dabei drei Aspekte ausmachen (vgl. [Abschnitt 1.1.2](#)): die Struktur (links, blau), die Funktion (mittig, gelb) und das Verhalten (rechts, grün). Die Schritte der Modellentwicklung sind kursiv benannt (vgl. [Abschnitt 1.3.1](#)). Dabei können die Einzelschritte (doppelte Linien) zusammengefasst werden zu Modellierung – Simulation – Evaluation (gestrichelte Linien). Kompetenz und Performanz (Schlangelinien) sind wichtige Eigenschaften dynamischer Modelle (vgl. [Abschnitt 1.1.3](#)).

## 2. Wissensfacetten von Bio-Modellen

„In itself, a computational model is just an ungrounded symbol system; no matter how closely it matches the properties of what is being modelled, it matches them only formally, with the mediation of an interpretation.“

---

(HARNAD 1994: *Levels of functional equivalence in reverse bioengineering*, S. 293)

In diesem Kapitel soll untersucht werden, welche Wissensfragmente für die Bildung und Verwendung von Bio-Modellen notwendig sind. Ausgehend von Überlegungen zur Interpretation von Bio-Modellen ([Abschnitt 2.1](#)) wird ein konzeptuelles Schema zur systematischen Beschreibung von Bio-Modellen entwickelt: Die *Wissensfacetten* betrachten ein Bio-Modell aus struktureller, funktionaler und verhaltensbezogener Sicht ([Abschnitt 2.2](#)). Dabei werden die resultierenden sechs Wissensfacetten von Bio-Modellen im Detail beschrieben. In [Abschnitt 2.3](#) werden die Wissensfacetten von Bio-Modellen schließlich mit anderen Konzeptualisierungen verglichen.

Bei der Entwicklung der Wissensfacetten spielen zwei grundlegende Faktoren eine Rolle: Die Dualität formaler Modelle (vgl. [Abschnitt 1.1.1](#)) und die Besonderheit dynamischer Modelle (vgl. [Abschnitt 1.1.2](#)). Ausgangspunkt der Entwicklung ist einerseits der menschliche Zugang zu Bio-Modellen: Wie wird ein Modell von einem Biologen verstanden und sinnvoll eingesetzt? Was bedeutet es für ihn? Welches vorhandene Wissen und welche zusätzlichen Informationen sind für das Verstehen und die Verwendung notwendig? Andererseits ist ein implementiertes Bio-Modell auch ein informatisches Artefakt, welches von entsprechender Software sinnvoll verarbeitet werden muss. Die dafür notwendige formale Semantik stellt eine andere Sichtweise auf die Interpretation von Bio-Modellen dar. Bei den Bio-Modellen handelt es sich um Modelle von dynamischen Vorgängen. Zur Bedeutung eines solchen dynamischen Modells gehört neben seiner Struktur auch sein Verhalten. Für die Interpretation eines Bio-Modells ist es außerdem notwendig, die Funktion anzugeben, die das Modell in einem wissenschaftlichen Prozess hat.

Dieses Kapitel liefert die konzeptuelle Grundlage für eine computergestützte Bio-Modellierung (vgl. [Abschnitt 1.3](#)), indem es relevante Wissensfragmente der systembiologischen Modellierung und Simulation identifiziert und systematisiert. Die anschließend notwendige Formalisierung der Wissensfacetten erfolgt in [Kapitel 4](#). Das hier vorgestellte konzeptuelle Schema stellt eine Weiterentwicklung von [KNÜPFER ET AL. 2006](#) dar und wurde in dieser Form in [KNÜPFER ET AL. 2013b](#) veröffentlicht.

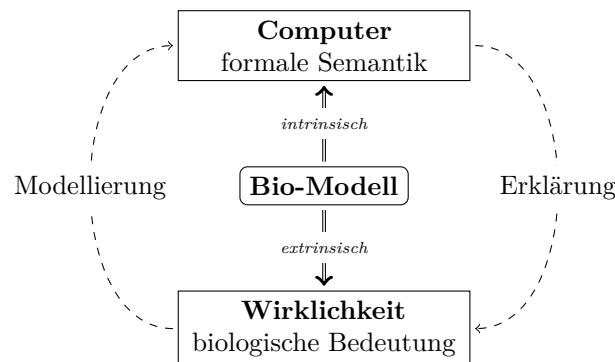


Abbildung 2.1.: Duale Interpretation von Bio-Modellen.

## 2.1. Interpretation von Bio-Modellen

In diesem Abschnitt wird untersucht, wie ein Bio-Modell interpretiert wird. Diese Interpretation erfolgt einerseits durch Software, die das Modell verarbeitet, z.B. für eine Simulation des dynamischen Verhaltens. Andererseits erfolgt die Interpretation durch Wissenschaftler, die das Modell für die Erklärung biologischer Phänomene benutzen. Mit dieser Dualität der Interpretation von Bio-Modellen setzt sich [Abschnitt 2.1.1](#) auseinander. Anschließend wird gezeigt, wie globale Metainformationen die Einbettung eines Bio-Modells in den wissenschaftlichen Kontext beschreiben ([Abschnitt 2.1.2](#)). Bei der Interpretation eines Bio-Modells bezüglich der modellierten biologisch-physikalischen Wirklichkeit gilt es, einige Besonderheiten zu beachten, die in [Abschnitt 2.1.3](#) diskutiert werden.

### 2.1.1. Duale Interpretation von Bio-Modellen

Wie bereits in [Abschnitt 1.1.1](#) dargestellt, codiert ein Modell einen bestimmten Ausschnitt der Wirklichkeit. Umgekehrt kann das Modell und sein Verhalten wieder in Wissen über diesen Ausschnitt der Wirklichkeit decodiert werden (vgl. [ROSEN 1991](#)). Das Modell wird also bezüglich eines Ausschnitts der biologischen Wirklichkeit interpretiert, indem es mit *biologischen Referenzen* versehen wird. Diese Art der Interpretation soll als **extrinsisch** bezeichnet werden. In der [SBML-Community](#) wird diesbezüglich aus offensichtlichen Gründen auch von „biological meaning“ gesprochen (z.B. [LE NOVÈRE 2010](#)).

Andererseits existiert eine formale Repräsentation des Modells. Diese *syntaktische* Struktur kann unabhängig von ihrer biologischen Bedeutung bezüglich des verwendeten Formalismus interpretiert werden. Damit wird seine *formale Semantik*<sup>1</sup> festgelegt, die hier als **intrinsisch** bezeichnet werden soll. Im [SBML-Kontext](#) wird auch von „model meaning“ gesprochen.

Der Begriff „biological meaning“ ist beschränkt auf biologische Modelle. Diese sind zwar Gegenstand vorliegender Arbeit, jedoch ist die duale Interpretation formaler Modelle eine generelle Eigenschaft. Weiterhin ist „model meaning“ zu allgemein und mehrdeutig. Daher werden hier stattdessen die Bezeichnungen „extrinsisch“ und „intrinsisch“ verwendet.

<sup>1</sup> „Formale Semantik“ hier im Sinne der Semantik von Programmiersprachen. Anliegen der vorliegenden Arbeit ist natürlich eine formale Semantik beider Interpretationen, d.h. eine Formalisierung sowohl der intrinsischen als auch der extrinsischen Bedeutung.

Ein Modell hat also eine duale Interpretation, welche in [Abbildung 2.1](#) veranschaulicht ist: Es ist ein formales Konstrukt, welches eine klar festgelegte formale Semantik besitzt. Dadurch kann das Modell in einem Computer effektiv simuliert bzw. mit dem Modell Berechnungen durchgeführt werden. Zum Bio-Modell wird es aber erst dadurch, dass es bezüglich der modellierten biologischen Wirklichkeit (re-)interpretiert wird. Erst dadurch können durch die Computersimulationen gewonnene Aussagen in *Erklärungen* biologischer Phänomene „decodiert“ ([ROSEN 1991](#)) werden. In diesem Zusammenhang spricht man oft davon, dass ein Modell bzw. ein Teil davon etwas „repräsentiert“.

Die duale Interpretation führt zu einer zweifachen Semantik von Modellen: der intrinsischen formalen Semantik und der extrinsischen biologischen Semantik. Es gibt jedoch einen qualitativen Unterschied zwischen beiden: Die intrinsische Semantik ist präzise spezifiziert und kann algorithmisch abgeleitet werden. Sie ist also im Sinne der *Physical Symbol System Hypothesis* ([NEWELL ET AL. 1976](#)) rein *syntaktisch* ableitbar. Dagegen wird die biologische Bedeutung vom Modellierer festgelegt, sie bezieht sich immer auf einen bestimmten Zweck. Aus diesem Grund wird auch von „intendierter Semantik“ gesprochen. Im Übrigen kann es durchaus sein, dass ein und dasselbe formale Konstrukt verschieden extrinsisch interpretiert wird. In der Sprache der *Physical Symbol System Hypothesis* legen die biologischen Referenzen die *semantischen* Aspekte eines Formalismus fest. Entsprechend der bekannten semiotischen Dreiteilung könnte man von Syntax (intrinsisch) und Semantik (extrinsisch) von Bio-Modellen sprechen. Sofort drängt sich die Frage nach der *pragmatischen* Sicht auf ein Bio-Modell auf. Die Pragmatik von Bio-Modellen wird im nächsten Abschnitt eine zentrale Rolle spielen.

### 2.1.2. Globale Metainformationen

Wie im letzten Abschnitt beschrieben, kann ein Bio-Modell sowohl formal als auch biologisch interpretiert werden. Neben dieser modellzentrierten Sicht kann ein Bio-Modell jedoch auch als ein Objekt im wissenschaftlichen Prozess aufgefasst werden, welches entwickelt, in einem computerverarbeitbaren Dateiformat implementiert und ausgetauscht wird. Dieser Abschnitt beschäftigt sich mit dem Wissen über ein Bio-Modell als Objekt im wissenschaftlichen Prozess. Solche Informationen über Informationen (in diesem Fall das implementierte Modell) werden als *Metainformationen* bezeichnet. Die globalen Metainformationen eines Bio-Modells beschreiben seine *Herkunft* und die *Publikation*, in der es veröffentlicht wurde, die *Implementierung* des Modells und den *Zugriff* auf das implementierte Modell sowie die *Beziehungen* zu anderen Modellen und die *Versionsgeschichte* des Modells. Im Detail stellen sich die globalen Metainformationen eines Bio-Modells, als [Facette M1](#) bis [Facette M3](#) bezeichnet (siehe auch [Abschnitt 2.2](#)), wie folgt dar:

#### Facette M1: Herkunft

Welche *Herkunft* hat das Modell? In welcher *Publikation* ist das Modell beschrieben?

Die Herkunft des Modells wird durch die Angabe der Autoren und des Datums seiner Veröffentlichung bestimmt. Die Publikation, in der das Modell beschrieben und spezifiziert ist, ist zu identifizieren.

### Facette M2: Implementierung

In welcher *Implementierung* liegt das Modell vor? Wie kann der *Zugriff* auf das implementierte Modell erfolgen?

Das in einer Publikation spezifizierte Modell muss in einem computerverarbeitbaren Dateiformat implementiert werden. Die Beschreibung der Implementierung umfasst das verwendete Dateiformat (z.B. [SBML](#), *Systems Biology Markup Language*, [HUCKA ET AL. 2003](#), [CellML](#), [CUELLAR ET AL. 2003](#), [MATLAB](#), *MATrix LABoratory*, <http://www.mathworks.de/>), die Verfasser und das Datum der Implementierung. Der Zugriff auf das implementierte Modell kann als Datei in einem lokalen Dateisystem, über eine [URL](#) (*Uniform Resource Locator*) im Internet oder durch Abfrage einer entsprechenden Modell-Datenbank erfolgen. Oft wird dem Modell ein Name gegeben, um es leichter referenzieren zu können. Es ist anzugeben, welche rechtlichen Bedingungen die Nutzung und Veränderung des Modells regeln (Lizenzvereinbarungen). Liegt das Modell als Datenbank-Eintrag vor, sind neben einer ID meist noch Informationen gegeben, die den Kurationsprozess betreffen, wie etwa Datum des Eintrags und der letzten Änderung, Revisionsnummer sowie Namen der Kuratoren.

### Facette M3: Beziehungen

Welche *Beziehungen* zu anderen Modellen sind ausgewiesen? Welche *Versionsgeschichte* weist das Modell auf?

Es gibt verschiedene Beziehungen, die ein Modell zu anderen Modellen aufweisen kann. Ein Modell kann eine Abstraktion, Konkretisierung oder Modifikation eines Modells sein. Bei der Integration und Kombination von Modellen gibt es Teil-Ganzes-Beziehungen zwischen Modellen. Außerdem kann ein Modell mit alternativen Modellen verglichen werden. Ein Modell durchläuft meist einen längeren Entwicklungsprozess. Eine detaillierte Beschreibung der Versionsgeschichte umfasst neben Datum und Urheber jeder Version Angaben zum Inhalt und den Gründen der Änderungen.

#### Beispiel 2.1 (Globale Metainformationen zu Modell 1 und Modell 2)

Beide Modelle stammen von John J. Tyson und wurden im August 1991 in [TYSON 1991](#) veröffentlicht (*Herkunft* und *Publikation*, [Facette M1](#)). Eine *Implementierung* ([Facette M2](#)) in [SBML](#) (Level 2, Version 4) liegt in der [BioModels Database](#) ([LE NOVÈRE ET AL. 2006](#)) vor. Der *Zugriff* ([Facette M2](#)) kann direkt über das Internet erfolgen:

Modell 1: <http://identifiers.org/biomodels.db/BIOMD0000000005>

Modell 2: <http://identifiers.org/biomodels.db/BIOMD0000000006> .

In der [BioModels Database](#) finden sich auch die entsprechenden Angaben zur Implementierung und zur Kuration. Hier nur die Wichtigsten:

Modell	Name	<a href="#">BioModels Database</a> -ID	Datum der Implementierung
Modell 1	Tyson1991_CellCycle_6var	BIOMD0000000005	08.02.2005
Modell 2	Tyson1991_CellCycle_2var	BIOMD0000000006	08.02.2005



Alle Modelle in der [BioModels Database](#) stehen unter der Lizenz „CC0 Public Domain Dedication“ ([CREATIVE COMMONS 2009](#)).

[TYSON 1991](#) gibt einige *Beziehungen* ([Facette M3](#)) seiner Modelle zu anderen Modellen an: Modell 1 ist aus zahlreichen anderen Modellen entwickelt worden (vgl. [TYSON 1991](#), S. 7329). Nach [TYSON 1991](#) (S. 7332) ist Modell 2 eine modifizierte Version des „Brüsselators“ ([NICOLIS ET AL. 1977](#)). Daneben besteht natürlich eine enge Beziehung zwischen den beiden Modellen: Modell 2 ist eine Abstraktion von Modell 1. [TYSON 1991](#) macht keine Angaben zur *Versionsgeschichte* ([Facette M3](#)) seiner Modelle. ■

### 2.1.3. Anmerkungen zur biologisch-physikalischen Interpretation

Bei der extrinsischen Interpretation von Bio-Modellen sind einige Besonderheiten zu beachten (vgl. [KNÜPFER ET AL. 2013b](#)). Diese resultieren letztendlich daraus, dass Modellierung immer zu einer Verkürzung führt (siehe [Abschnitt 1.1](#)) und es verschiedene Möglichkeiten gibt, ein biologisches Phänomen durch ein Modell zu beschreiben. Die Auswahl relevanter Merkmale und einer geeigneten Repräsentation hängen von den Intentionen des Modellierers und nicht zuletzt von dessen kreativen Fähigkeiten ab. Neben diesen Besonderheiten der biologischen Interpretation werden Hinweise zur numerischen Repräsentation physikalischer Größen gegeben.

#### Biologische Bedeutung

Bei der Angabe der biologischen Bedeutung sind folgende Hinweise zu beachten:

1. Im Allgemeinen wird die extrinsische Interpretation eines Bio-Modells nur partiell erfolgen können. Es wird meist Aspekte des Modells geben, für die keine biologischen Gegenstücke existieren, die also nicht extrinsisch interpretiert werden können. Die Forderung nach der Kompetenz eines Bio-Modells bedeutet aber, dass zumindest wichtige Aspekte eines Bio-Modells eine extrinsische Interpretation besitzen (vgl. [Abschnitt 1.1.3](#)). Ein Bio-Modell, welches kein konkretes biologisches System repräsentiert, wäre zumindest aus biologischer Sicht „bedeutungslos“.
2. Auch wenn eine extrinsische Interpretation eines Modellaspekts existiert, kann diese Interpretation recht schwierig erfassbar sein: Es gibt Modelle, die bestimmte Aspekte der biologischen Wirklichkeit auf unkonventionelle und nicht offensichtliche Art und Weise darstellen. So ist es z.B. schwierig, den ersten Summanden  $k_4(v - u)(\alpha + u^2)$  der [Gleichung 1.17](#) von Modell 2 extrinsisch zu interpretieren, obwohl natürlich extrinsische Interpretationen für die einzelnen Variablen existieren. Je intuitiver ein Bio-Modell jedoch unsere Wahrnehmung der biologischen Wirklichkeit widerspiegelt, umso besser wird es die modellierten Phänomene erklären und schlussendlich zu einem besseren Verständnis der biologischen Vorgänge beitragen.
3. Die extrinsische Interpretation eines Bio-Modells hängt stark von den Intentionen des Modellierers ab. So kann etwa ein mathematisches Konstrukt – wie das Gesetz für exponentielles Wachstum  $\dot{x} = \alpha x, \alpha > 0$  (vgl. [Abschnitt 1.1.1](#)) – für die Modellierung verschiedener biologischer Phänomene verwendet werden, also unterschiedliche biologische Bedeutung besitzen.

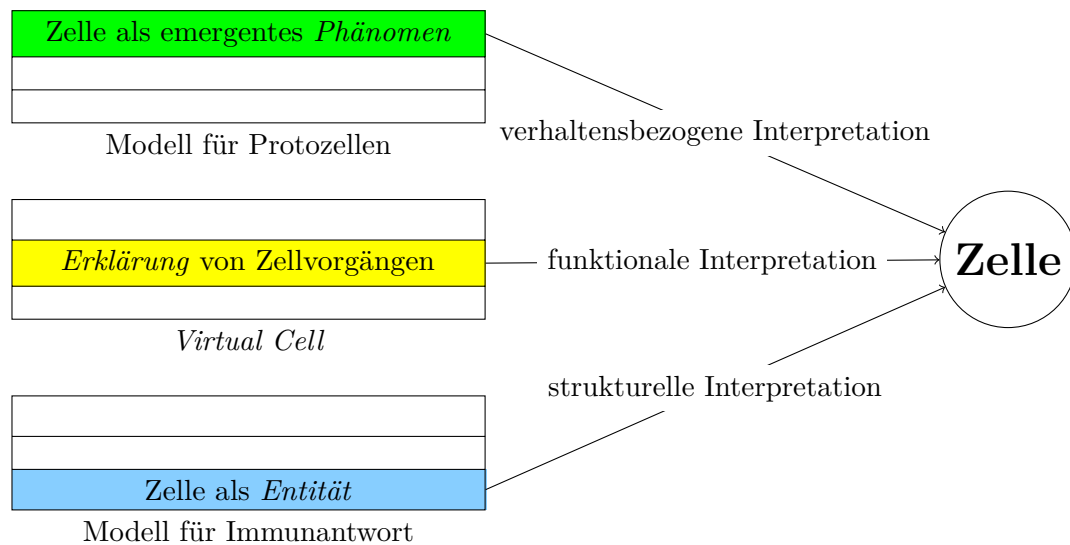


Abbildung 2.2.: Verschiedene Betrachtungsweisen einer „Zelle“ in unterschiedlichen Modellen: Zelle als Verhalten in einem Modell für Protozellen (ONO ET AL. 1999), Zelle als Funktion in der *Virtual Cell* (SCHAFF ET AL. 1997) und Zelle als Struktur in einem Modell für Immunantwort (NOWAK ET AL. 1996).

4. Bekannte biologische Strukturen, wie z.B. die Zelle, müssen nicht zwangsläufig strukturell als Modellkomponente in einem Bio-Modell repräsentiert sein. [Abbildung 2.2](#) illustriert den Umstand, dass eine Zelle sowohl strukturell als Modellkomponente (Entität) auftreten kann, sich aber auch im Verhalten des Modells zeigen kann, etwa als emergentes Phänomen bei der Simulation eines Modells. Darüber hinaus kann eine Zelle auch der Untersuchungsgegenstand eines Modells sein und somit zur Beschreibung der Funktion des Modells gehören. Biologische Strukturen besitzen also auch funktionale und verhaltensbezogene Aspekte. Diese drei Betrachtungsweisen von Bio-Modellen (siehe [Abschnitt 2.2](#)) sollten also keinesfalls als vollständig unabhängig voneinander angesehen werden. Im Gegenteil: Die Wissensfacetten bieten unterschiedliche Sichtweisen auf ein Bio-Modell und auf die durch das Bio-Modell repräsentierte unteilbare Wirklichkeit.
5. Die extrinsische Interpretation eines Bio-Modells wird durch die zugehörige Publikation gegeben, die in ein sehr spezifisches Problemfeld eingebunden ist. Für das Verstehen der biologischen Bedeutung eines Bio-Modells ist also spezifisches Fachwissen und ein großes Maß an biologischem Hintergrundwissen notwendig. Darüber hinaus muss für die Zuordnung der biologischen Bedeutung zu Modellaspekten auch auf detailliertes Wissen über den verwendeten Formalismus, die eingesetzte Simulationssoftware sowie auf allgemeines mathematisches Wissen zurückgegriffen werden.

### Physikalische Dimensionen und Maßeinheiten

Bei der numerischen Simulation von Modellen treten verschiedene intrinsische Zahlenwerte auf, die bestimmte quantitative physikalische Größen repräsentieren. Für die Durchführung

einer Simulation müssen aktuelle Parameter und ein initialer Zustand gegeben sein, die Simulation liefert numerische Daten als Ergebnis. Jede repräsentierte physikalische Größe besitzt eine entsprechende Dimension, ihre Werte werden bezüglich geeigneter Maßeinheiten angegeben. Bei der Codierung aktueller Parameter und initialer Zustände in numerische Werte fallen die Maßeinheiten weg. Bei der Decodierung der Ergebnisse einer Simulation müssen den numerischen Daten in geeigneter Weise Maßeinheiten zugeordnet werden. Diese müssen zum einen den Dimensionen der repräsentierten physikalischen Größen gerecht werden und zum anderen mit der Codierung der aktuellen Parameter und initialen Zustände sowie mit den mathematischen Gleichungen des Modells verträglich sein. Dabei müssen die aus der Physik bekannten Rechenregeln für Maßeinheiten beachtet und Maßeinheiten eventuell umgerechnet werden. Am besten lassen sich diese Forderungen am Beispiel erklären:

### Beispiel 2.2 (Maßeinheiten in Modell 1)

Im Folgenden wird [Gleichung 1.6](#) (siehe [Seite 33](#)) von [Modell 1](#) betrachtet:

$$\frac{d[C2]}{dt} = k_6[M] - k_8[\sim P][C2] + k_9[CP] . \quad (1.6)$$

Für eine zeitliche Simulation müssen zunächst für die Parameter  $k_6$ ,  $k_8$  und  $k_9$  aktuelle Werte gegeben und geeignet numerisch codiert werden. Außerdem sind die initialen Konzentrationen  $[M]$ ,  $[\sim P]$ ,  $[C2]$  und  $[CP]$  bereitzustellen. Physikalisch gesehen müssen alle drei Summanden von [Gleichung 1.6](#) dieselbe Maßeinheit besitzen und von der Dimension „Konzentration je Zeit“ (im internationalen Einheitensystem  $N/L^3/T$ , also Stoffmenge je Volumen und Zeit) sein. Insbesondere heißt dies, dass die initialen Konzentrationen alle die gleiche Maßeinheit aufweisen müssen. Die Parameter  $k_6$  und  $k_9$  haben die Dimension  $1/T$  und  $k_8$  hat die Dimension  $L^3/N/T$ . Die aktuellen Parameter müssen in entsprechenden Maßeinheiten gegeben sein. Die Werte in [Tabelle 1.2](#) erfüllen diese Bedingungen. Die Ergebnisse einer mit diesen Werten durchgeführten zeitlichen Simulation sind wie folgt zu interpretieren: Die Simulationszeit  $t$  hat die Maßeinheit min und die Konzentration  $[C2]$  hat die Maßeinheit  $\text{mol} \cdot \text{l}^{-1}$ . ■

Besonders anschaulich wird diese Abhängigkeit der Maßeinheiten decodierter Simulationsergebnisse von den Maßeinheiten der codierten Eingabedaten bei Betrachtung einer gewissen „Invarianz“ der Simulationsergebnisse. So kann etwa in [COPASI](#) (*C*OMplex *P*ATHway *S*IMulator, [HOOPS ET AL. 2006](#)) die Maßeinheit der Simulationszeit frei gewählt werden. Wird diese für ein Modell z.B. von min auf s geändert, verändert sich numerisch gesehen überhaupt nichts: die Parameter und initialen Konzentrationen haben dieselben Werte und die Simulation einer Zeitreihe liefert numerisch gleiche Werte. Die grafische Darstellung des Zeitverlaufs sieht bis auf die Abszissenbeschriftung identisch aus.

## 2.2. Drei Betrachtungsweisen von Bio-Modellen

Bio-Modelle konstituieren Beziehungen zwischen einem formalem System (dem „Modell“ im herkömmlichen Sinne) und einer spezifischen biologischen Wirklichkeit (dem modellierten Phänomen). Im [Abschnitt 2.1.1](#) wurde gezeigt, wie die Interpretation eines Bio-Modells entsprechend dieser beiden Sichtweisen erfolgt.

Die folgende Dreiteilung trägt den Aussagen in [Abschnitt 1.1.2](#) Rechnung und entspricht den Ideen der funktionalen Modellierung (siehe [Abschnitt 2.3](#)). Für Bio-Modelle lässt sich diese Dreiteilung wie folgt motivieren:

Offensichtlich hat ein Modell eine (intrinsische) *Struktur* – das formale System kann als Netzwerk (vgl. [Abschnitt 1.1.2](#)), d.h. als statische Ansammlung von Komponenten angesehen werden, die miteinander auf bestimmte Art verknüpft sind. Da es sich um ein Kompetenz-Modell handelt (vergleiche [Abschnitt 1.1](#)), tragen Modellkomponenten eine Bedeutung bezüglich dem modellierten System, d.h. die intrinsische Struktur eines Bio-Modells korrespondiert mit der extrinsischen Struktur des modellierten biologischen Systems. Wie in [Kapitel 1](#) dargelegt, ist ein Bio-Modell ein dynamisches Modell, d.h. es wird verwendet, um das dynamische *Verhalten* eines biologischen Systems zu simulieren und zu analysieren.

Die Struktur eines Modells legt das mögliche dynamische Verhalten fest. Im Allgemeinen ist der Zustandsraum eines formalen Systems unendlich groß oder bei numerischen Approximationen zumindest gigantisch (die normale IEEE 754-Gleitkommazahl einfacher Genauigkeit erlaubt bereits mehr als  $10^9$  unterschiedliche Werte). Damit besitzt z.B. bereits ein formales System mit nur einer einzigen Zustandsvariablen vom Typ `float` eine unüberschaubare Anzahl konkreter zeitlicher Verläufe, jeweils eine für jeden möglichen Startzustand. Diese lassen sich meist in eine kleine Zahl von Verhaltensklassen einteilen. Eine solche Klassifizierung von Verhaltensweisen kann man als die Haupttätigkeit der Theorie dynamischer Systeme ansehen (vgl. [Abschnitt 5.1](#)). Desweiteren sind die meisten Modelle parametrisiert, d.h. bestimmte Werte können frei vom Benutzer gewählt werden. Die Wahl plausibler Parameterwerte und sinnvoller Startzustände hängt von der biologischen Fragestellung ab. Ein Modell kann also nur dann sinnvoll für die Simulation des Verhaltens eines biologischen Systems eingesetzt werden, wenn man etwas über seine *Funktion* sagt.

Jede dieser drei Betrachtungsweisen hat sowohl extrinsische als auch intrinsische Aspekte. Zusammen ergeben sich also sechs *Wissensfacetten* von Bio-Modellen. Die Wissensfacetten werden durch den sogenannten „Wissensdiamant“ illustriert (siehe [Abbildung 2.7](#) auf [Seite 80](#)).

Die hier vorgestellte Konzeptualisierung der Wissensfacetten von Bio-Modellen geht zurück auf sprachanalytische Betrachtungen der Präsentation und der Verwendung von Bio-Modellen. Die Wissensfacetten systematisieren das Wissen über ein Bio-Modell, dessen Verwendung und das resultierende Verhalten. Da der Ausgangspunkt der hier entwickelten Konzeptualisierung die Interpretation eines vorhandenen Bio-Modells ist, könnte auch von „Bedeutungsfacetten“ gesprochen werden. Die hier verwendete Bezeichnung „Wissensfacetten“ scheint jedoch angemessener.

In den folgenden Unterabschnitten wird auf die einzelnen Betrachtungsweisen von Bio-Modellen im Detail eingegangen und diese am Beispiel illustriert. Dabei erfolgt für die Struktur ([Abschnitt 2.2.1](#)), die Funktion ([Abschnitt 2.2.2](#)) und das Verhalten ([Abschnitt 2.2.3](#)) von Bio-Modellen zunächst jeweils eine zusammenfassende Darstellung. Anschließend werden jeweils durch drei Unterfacetten die intrinsischen und extrinsischen Aspekte von Bio-Modellen konkretisiert. Zur Illustration wird abschließend jeweils das Wissen über die TYSON-Modelle bezüglich der jeweiligen Betrachtungsweise skizziert.

Zur späteren Referenzierung werden die Unterfacetten nummeriert. Dabei kennzeichnet ein vorangestellter Buchstabe die Betrachtungsweise. Die Unterfacetten sind entsprechend:

Facette S1 bis Facette S3 (Struktur), Facette F1 bis Facette F3 (Funktion) und Facette B1 bis Facette B3 (Verhalten, „B“ vom englischen „behaviour“). Hinzu kommen die Unterfacetten der globalen Metainformationen (siehe Abschnitt 2.1.2): Facette M1 bis Facette M3.

### 2.2.1. Struktur eines Bio-Modells

Ein dynamisches Bio-Modell beschreibt zeitliche Veränderungen. Die *Struktur* bezieht sich auf diejenigen Aspekte eines Modells, die sich nicht verändern. Das *formale System* stellt einerseits die Struktur des Modells in Form eines Netzwerks dar und ermöglicht seine Kompetenz. Andererseits legt das *dynamische System* fest, wie sich der Zustand des formalen Systems zeitlich ändern kann. Damit erst kann das Modell ein Performanz-Modell bezüglich der zu erklärenden biologischen Phänomene werden (siehe Abschnitt 1.1.3).

Ganz allgemein kann die Struktur des formalen Systems immer als eine Menge von *Entitäten*<sup>2</sup> und *Relationen* zwischen diesen dargestellt werden:

„Every representational formalism I know [...], is based on the idea of individual entities and relationships between them.“ (HAYES 1985a, S. 72)

Noch allgemeiner kann man jeden (syntaktischen) sprachlichen Ausdruck als Menge von Entitäten (Zeichen aus einem endlichen Alphabet) und Relationen zwischen diesen (Anordnung der Zeichen im Ausdruck) auffassen, also als eine Struktur. So stellen zum Beispiel Differentialgleichungen auch eine Struktur dar, die Struktur der symbolischen Beschreibung eines dynamischen Systems.

Jede Entität ist durch spezifische *Attribute* charakterisiert, die zeitlichen Veränderungen unterworfen sein können. Die Gesamtheit der Attribute der Entitäten konstituieren den Gesamtzustand des dynamischen Systems. Die Relationen beschreiben Abhängigkeiten zwischen entsprechenden Entitäts-Attributen. Das dynamische System legt das *Programm* fest, nach dem sich der Systemzustand zeitlich ändert. Dabei sind die durch die Relationen beschriebenen Abhängigkeiten notwendige Bedingungen für die durch das dynamische System bewirkten Zustandsänderungen: Bewirkt eine Änderung eines Attributes die Änderung eines anderen Attributes, so muss es eine Relation zwischen den entsprechenden Entitäten geben.

Um das Modell für Erklärungen biologischer Phänomene verwenden zu können, muss die intrinsische Struktur des formalen Systems auf die Struktur des untersuchten *biologischen Systems* abgebildet werden, d.h. auf die relevanten biologischen *Objekte* und *Interaktionen* (vgl. Abschnitt 1.2). Dabei werden den intrinsischen Attributen geeignete *physikalische Größen* der entsprechenden Objekte zugeordnet. Änderungen im biologischen System werden durch *Mechanismen* beschrieben, welche durch das Programm des Modells widerspiegelt werden. Das biologische System wird dabei auf einer festgelegten *epistemischen Ebene* betrachtet.

Die strukturelle Interpretation eines Bio-Modells wird in der Regel „rekursiv“ sein. Sie folgt der Teil-Ganzes-Beziehung der Modellkomponenten (rekursiver Aufbau). Das Modell als Ganzes bzw. das modellierte System kann dabei als die oberste Entität in der partonomischen Hierarchie aufgefasst werden.

---

<sup>2</sup>„Entität“ soll hier nicht im ontologischen Sinne verstanden werden, sondern in der in der Informatik üblichen allgemeineren Lesart als konzeptuelle Einheit

Die Struktur eines Bio-Modells kann im Einzelnen durch folgende Fragen charakterisiert werden:

### Facette S1: System

**Intrinsisch:** Welches *formale System* wird durch das Modell spezifiziert? Welches *dynamische System* beschreibt Zustandsänderungen des formalen Systems?

Für die Beschreibung des formalen Systems muss zunächst der verwendete Formalismus festgelegt werden. Oft wird das formale System in Form einer entsprechenden grafischen Darstellung präsentiert, meist als gerichteter Graph oder Hypergraph. Eine Beschreibung des dynamischen Systems umfasst den verwendeten Kalkül und damit auch die Festlegung grundlegender Eigenschaften von Zustandsänderungen, wie Art der Zeitentwicklung (diskret oder kontinuierlich), Beachtung von Räumlichkeit und Art des Systemverhaltens (deterministisch oder stochastisch). In [Abschnitt 1.2](#) findet sich eine Beschreibung wichtiger Formalismen und Kalküle, die in der Systembiologie Verwendung finden.

**Extrinsisch:** Welches *biologische System* korrespondiert mit dem formalen System des Modells? Auf welcher *epistemischen Ebene* wird das biologische System betrachtet?

Für die Verwendung des Modells zur Erklärung biologischer Phänomene muss zunächst angegeben werden, welches biologische System die zu erklärenden Phänomene aufweist. Mit „biologisches System“ kann hier auch eine Klasse von konkreten biologischen Systeminstanzen gemeint sein. So umfasst etwa das System „Zelle“ die Klasse aller konkreten Zellen. Die Bestimmung des modellierten biologischen Systems hat so genau wie möglich zu erfolgen, etwa durch Angabe der taxonomischen Einheit, der Zell- und Gewebetypen sowie des Entwicklungsstadiums. Das modellierte biologische System wird für die angestrebte Erklärung auf einer festgelegten epistemischen Ebene betrachtet. Die epistemische Ebene ist durch den Systemtyp, die Art der betrachteten biologischen Objekte (z.B. Moleküle, Gene, Zellen) und deren Interaktionsformen (z.B. chemische Reaktionen, Umwandlungen, Diffusion, Kompartimentierung) sowie die räumliche und zeitliche Granularität gekennzeichnet. Außerdem stellt die epistemische Ebene die elementaren Mechanismen bereit, die dynamische Änderungen realisieren (z.B. kinetische Gesetze, Reaktionsschritte, Bindungen, Aktivierungen). In [Abschnitt 1.2](#) sind typische Systemtypen und elementare Mechanismen aufgeführt, die in der Systembiologie benutzt werden.

### Facette S2: Entitäten

**Intrinsisch:** Aus welchen *Entitäten* ist das formale System aufgebaut? Welche *Attribute* der Entitäten beschreiben den Systemzustand?

Das formale System ist aus Entitäten aufgebaut. Diese müssen aufgelistet und jeweils einem Typ zugeordnet werden. Jede Entität muss dabei eine eindeutige Bezeichnung besitzen, um sie referenzieren zu können. Den Entitäten werden Attribute zugeordnet, welche den Zustand des dynamischen Systems bestimmen. Für jedes Attribut muss der Wertebereich festgelegt werden. Die Attribute finden sich als typisierte Variablen im dynamischen System wieder.



**Extrinsisch:** Welche biologischen *Objekte* korrespondieren mit den Entitäten des Modells? Welche *physikalischen Größen* korrespondieren mit den Attributen der Modell-Entitäten?

Für mechanistische Erklärungen biologischer Phänomene müssen die Entitäten des formalen Systems auf Objekte des biologischen Systems abgebildet werden. Dafür müssen zunächst die Objekte des biologischen Systems festgelegt und beschrieben werden. Die Attribute der Modell-Entitäten repräsentieren physikalische Größen der zugehörigen biologischen Objekte. Als physikalische Größen kommen prinzipiell alle messbaren Eigenschaften eines biologischen Objektes in Frage, wie etwa Ort, Masse, Anzahl, Konzentration, Aktivität.

### Facette S3: Relationen

**Intrinsisch:** Welche *Relationen* bestehen zwischen den Entitäten? Welches *Programm* steuert die Attributsänderung von Entitäten, zwischen denen eine Relation besteht?

Zwischen Entitäten des formalen Systems können Relationen bestehen. Neben der Aufzählung der beteiligten Entitäten bekommt jede Relation eine eindeutige Bezeichnung. Die Relationen stellen notwendige Bedingungen für Abhängigkeiten zwischen den zugehörigen Attributen im dynamischen System dar. Das Programm legt fest, wie sich der Zustand des dynamischen Systems ändert. Je nach verwendetem Kalkül besteht das Programm etwa aus Regeln, Gleichungen oder Anweisungen. Die Relationen spiegeln sich im Programm oft dadurch wider, dass die entsprechenden Attribute der involvierten Entitäten auf verschiedenen Seiten einer Regel, Gleichung oder Zuweisung stehen. Das Programm besitzt in der Regel formale Parameter. Für diese muss festgelegt werden, welchen Wertebereich sie annehmen können.

**Extrinsisch:** Welche biologischen *Interaktionen* korrespondieren mit den Relationen? Welche biologischen *Mechanismen* realisieren die Interaktionen und korrespondieren mit dem Programm?

Die Relationen zwischen Entitäten des formalen Systems müssen auf Interaktionen zwischen entsprechenden Objekten des biologischen Systems abgebildet werden. Die Interaktionen führen zu dynamischen Änderungen im biologischen System. Wie diese Änderungen im Detail aussehen, wird durch Mechanismen beschrieben. Oft ist es möglich, eine Beziehung zwischen einem Teil des dynamischen Systems und einem Mechanismus herzustellen: z.B. zwischen formalen Parametern und kinetischen Ratenkonstanten, zwischen Exponenten und Reaktionsordnungen und zwischen Termen und Geschwindigkeitsgesetzen, wie der bekannten Michaelis-Menten-Gleichung.

### Beispiel 2.3 (Struktur von Modell 1)

Die intrinsische Struktur von Modell 1 ist durch das in [Abbildung 2.3](#) als Netzwerk dargestellte *formale System* ([Facette S1](#)) gegeben. Dieses Netzwerk korrespondiert mit dem extrinsischen Reaktionsnetzwerk (siehe [Abbildung 1.3](#)). Das *dynamische System* ([Facette S1](#)) von Modell 1 ist durch [Gleichung 1.6](#) bis [Gleichung 1.11](#) spezifiziert. Der verwendete Kalkül ist ein gekoppeltes Differentialgleichungssystem (siehe [Abschnitt 1.2](#)) in einer gemeinsamen unabhängigen Variablen  $t$ . Die *Entitäten* und *Relationen* ([Facette S3](#)) sind neben der grafischen Darstellung ([Abbildung 2.3](#)) noch einmal tabellarisch zusammengefasst in [Tabelle 2.1](#). Jede Entität  $x$  besitzt genau ein zugehöriges *Attribut* ([Facette S2](#)), welches durch die Schreibweise  $[x]$  symbolisiert wird. Die Attribute bestimmen den Zustand des

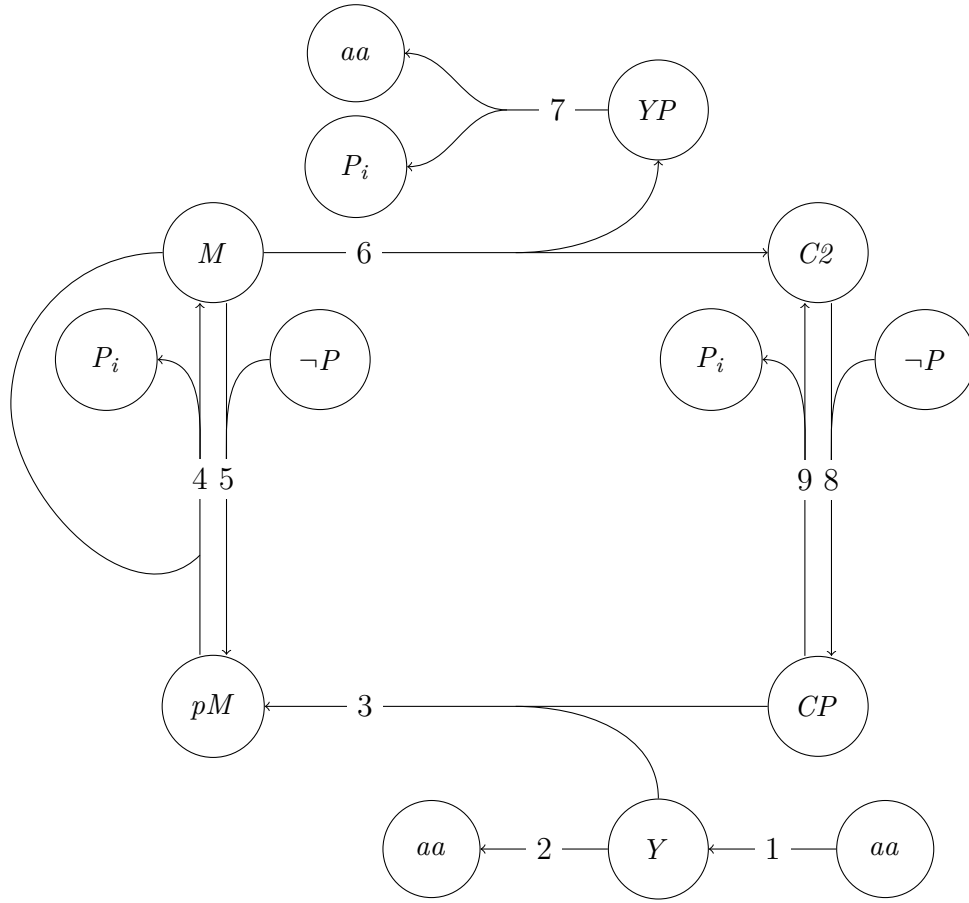


Abbildung 2.3.: Formales System ([Facette S1](#)) von [Modell 1](#) als Hypergraph dargestellt: Die Knoten repräsentieren die Entitäten ([Facette S2](#)), die Hyperkanten die Relationen ([Facette S3](#)). Die Entitäten  $aa$ ,  $\sim P$  und  $P_i$  sind aus Gründen der Übersichtlichkeit jeweils durch mehrere Knoten repräsentiert. Die Kantenlabel geben die Nummern der Relationen entsprechend den korrespondierenden Reaktionen aus [Abbildung 1.3](#) an. Für die einzelnen Entitäten und Relationen siehe auch [Tabelle 2.1](#).

dynamischen Systems und tauchen als Variablen im Differentialgleichungssystem auf. Das zugehörige *Programm* ([Facette S3](#)) ist durch die Differentialgleichungen realisiert. Jede Relation zwischen Entitäten findet sich als Summand in den Differentialgleichungen wieder, deren abhängige Variablen die Attribute der involvierten Entitäten sind (vgl. [Abschnitt 1.2](#)). Z.B. findet sich für die Relation 6 ([Abbildung 2.3](#)) von  $M$  zu  $C2$  und  $YP$  der Term  $k_6[M]$  in [Gleichung 1.6](#) und [Gleichung 1.11](#) sowie der Term  $-k_6[M]$  in [Gleichung 1.9](#). Dabei stellt  $k_6$  einen formalen Parameter dar. Im Netzwerk sieht man noch weitere Entitäten ( $aa$ ,  $\sim P$  und  $P_i$ ), die einen Spezialfall darstellen: die zugehörigen Attribute  $[aa]$ ,  $[\sim P]$  und  $[P_i]$  werden im Modell als Konstanten behandelt. Aus diesem Grund gibt es auch keine Differentialgleichung, die diese Attribute als abhängige Variable führen. Das Modell kann



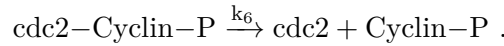
aber leicht um die entsprechenden Gleichungen ergänzt werden:

$$\frac{d[aa]}{dt} = 0 \quad (2.1)$$

$$\frac{d[\sim P]}{dt} = 0 \quad (2.2)$$

$$\frac{d[P_i]}{dt} = 0 . \quad (2.3)$$

Extrinsisch beschreibt **Modell 1** das *biologische System* ([Facette S1](#)) der Bildung und Aktivierung von **MPF** (*Maturation Promoting Factor*) (vgl. [Abschnitt 1.2.3](#)). Das biologische System wird auf der *epistemischen Ebene* ([Facette S1](#)) eines Reaktionsnetzwerks von Protein-Protein-Interaktionen betrachtet, in welchen katalytische Reaktionen zwischen Proteinen deren Konzentrationen ändern (siehe [Abschnitt 1.2](#)). Die Entitäten des **Modells 1** repräsentieren Proteine (biologische *Objekte*, [Facette S2](#)). Z.B. repräsentiert *C2* das Protein **CDK1** (*Cyclin-dependent Kinase 1*) (*cdc2* in [Abbildung 1.3](#)). Die zugehörigen Attribute repräsentieren die entsprechenden Konzentrationen (*physikalische Größe*, [Facette S2](#)). Den Relationen von **Modell 1** sind als extrinsische *Interaktionen* ([Facette S3](#)) entsprechende katalytische Reaktionen zugeordnet, z.B. stellt Relation 6 ([Abbildung 2.3](#)) die Dissoziation des aktiven **MPF** (*cdc2*–*Cyclin*–*P*) in **CDK1** (*cdc2*) und phosphoryliertes Cyclin (*Cyclin*–*P*) mit der Reaktionskonstante  $k_6$  dar (vgl. [Abbildung 1.3](#)):



Den Reaktionen von **Modell 1** liegen verschiedene *Mechanismen* ([Facette S3](#)) zugrunde (Nummern der Reaktionen wie in [Abbildung 1.3](#)): Zerfallsreaktionen erster Ordnung (Reaktionen 2, 6 und 7), Bildungsreaktionen erster und zweiter Ordnung (Reaktionen 1, 3 und 5), Gleichgewichtsreaktion nach dem Massenwirkungsgesetz (Reaktionen 8 und 9) und eine auto-katalytische Zerfallsreaktion (Reaktion 4). Die Zuordnungen zwischen der intrinsischen und der extrinsischen Struktur von **Modell 1** sind in [Tabelle 2.1](#) auf [Seite 66](#) zusammengefasst. ■

#### Beispiel 2.4 (Struktur von Modell 2)

In [TYSON 1991](#) wird für **Modell 2** lediglich ein *dynamisches System* ([Facette S1](#)) angegeben ([Gleichung 1.17](#) und [Gleichung 1.18](#)). Ohne Angabe einer Netzwerkstruktur des formalen Systems ist **Modell 2** kein Kompetenz-Modell. Die intendierte Verwendung von **Modell 2** in [TYSON 1991](#) ist dementsprechend auch nicht die Erklärung eines biologischen Phänomens, sondern die Analyse des dynamischen Verhaltens von **Modell 1**. **Modell 2** ist somit ein Modell *für* **Modell 1**. Die Verbindung zwischen beiden Modellen aus struktureller Sicht ist durch [Gleichung 1.15](#) bis [Gleichung 1.19](#) gegeben.

Die intrinsische Struktur von **Modell 2** beschränkt sich auf die Beschreibung des dynamischen Systems. Dieses besteht aus den *Attributen* ([Facette S2](#))  $u$  und  $v$  sowie den beiden Differentialgleichungen [Gleichung 1.17](#) und [Gleichung 1.18](#) als *Programm* ([Facette S3](#)). Der verwendete Kalkül ist wie in [Beispiel 2.3](#) ein gekoppeltes Differentialgleichungssystem.

Die extrinsische biologische Interpretation von **Modell 2** ist durch das Fehlen einer intrinsischen Netzwerkstruktur des formalen Systems nur teilweise möglich und kann nur über die Vermittlung durch **Modell 1** erfolgen. **Modell 2** beschreibt dasselbe *biologische*

Entitäten (Facette S2)			
Entität	Attribut	Objekt	physikalische Größe
$C2$	$[C2]$	CDK1	Konzentration
$CP$	$[CP]$	phosphoryliertes CDK1	Konzentration
$Y$	$[Y]$	Cyclin	Konzentration
$YP$	$[YP]$	phosphoryliertes Cyclin	Konzentration
$M$	$[M]$	aktiver MPF	Konzentration
$pM$	$[pM]$	inaktiver MPF	Konzentration
$\sim P$	$[\sim P]$	ATP	Konzentration
$P_i$	$[P_i]$	anorganische Phosphate	Konzentration
$aa$	$[aa]$	Aminosäuren	Konzentration

Relationen (Facette S3)			
Relation	Term	Interaktion	Mechanismus
1	$k_1[aa]$	Bildung von Cyclin	Bildungsreaktion 1. Ord.
2	$k_2[Y]$	Zerfall von Cyclin	Zerfallsreaktion 1. Ord.
3	$k_3[CP][Y]$	Dimerisation ph. CDK1 und Cyclin	Bildungsreaktion 2. Ord.
4	$[pM]F([M])$	Aktivierung von MPF	Zerfallsreaktion auto-kat.
5	$k_5[\sim P][M]$	Deaktivierung von MPF	Bildungsreaktion 2. Ord.
6	$k_6[M]$	Dissoziation von MPF	Zerfallsreaktion 1. Ord.
7	$k_7[YP]$	Zerfall von ph. Cyclin	Zerfallsreaktion 1. Ord.
8	$k_8[\sim P][C2]$	Phosphorylierung von CDK1	Bildungsreaktion 2. Ord.
9	$k_9[CP]$	Dephosphorylierung von ph. CDK1	Zerfallsreaktion 1. Ord.

Tabelle 2.1.: Strukturelle Facetten von Modell 1: Zuordnung intrinsischer und extrinsischer Strukturelemente. Für die vollständigen Namen der chemischen Spezies (Objekte) siehe Tabelle 1.1. Die Nummern der Relationen entsprechen denen in Abbildung 2.3 sowie den Nummern der korrespondierenden Reaktionen in Abbildung 1.3. Abkürzungen: kat.=katalytisch, Ord.=Ordnung, ph.=phosphoryliertem.

System (Facette S1) wie Modell 1 (siehe Beispiel 2.3). Gleichung 1.15 und Gleichung 1.16 erlauben eine Zuordnung von den Attributen zu *physikalischen Größen* (Facette S2) (Konzentrationen) entsprechender biologischer *Objekte* (Facette S2): Den Attributen  $u$  und  $v$  entsprechen Konzentrationen, die auf die Konzentration des gesamten CDK1 (repräsentiert durch  $[CT]$ ) bezogen sind. Dabei wird in TYSON 1991 angenommen, dass  $[CT] = \text{const.}$  gilt.  $u$  entspricht der relativen Konzentration des aktiven MPF,  $v$  der relativen Konzentration des Gemisches aus aktivem MPF, inaktivem MPF und Cyclin. Die Zuordnung des Programms von Modell 2 zu biologischen *Mechanismen* (Facette S3) ist aufgrund des Fehlens einer Darstellung des formalen Systems als Netzwerk unmöglich. Lediglich die Konstanten  $k_1$ ,  $k_4$ ,  $k'_4$  und  $k_6$  können denselben kinetischen Konstanten wie in Beispiel 2.3 zugeordnet werden. ■

### 2.2.2. Funktion eines Bio-Modells

Die Funktion eines Bio-Modells beschreibt die Verwendung der Modell-Struktur in Simulationen zur Untersuchung des Modell-Verhaltens. Funktion kann sowohl teleologisch als auch pragmatisch aufgefasst werden (vgl. den Abschnitt über funktionale Modellierung in [Abschnitt 2.3](#)). Bezogen auf die Simulation von Bio-Modellen werden mit der Funktion das Warum und das Wie beantwortet. Es ist zusätzlich notwendig, anzugeben, was simuliert werden soll. Die Festlegung der zu simulierenden Modellinstanz und eines ausgezeichneten Zustands des zugehörigen dynamischen Systems soll „konditionale Funktion“ genannt werden. Die konditionale Funktion charakterisiert den Kontext der Modellverwendung.

Die teleologische Funktion legt die *intendierte Verwendung* des Modells in Simulationen fest. Die intendierte Verwendung spiegelt die *biologischen Fragen* wider, die mit Hilfe des Modells untersucht werden sollen. Dabei beschränken *Constraints* die möglichen Verhaltensweisen des dynamischen Systems und tragen so den *Annahmen* Rechnung, die den Rahmen zur Beantwortung der biologischen Fragen setzen. Ein Modell muss zunächst vollständig *instantiiert* werden (konditionale Funktion), um in Simulationen verwendet werden zu können: Die Formal-Parameter müssen konkrete Werte erhalten und die Zustandsvariablen des dynamischen Systems müssen *initialisiert* werden. Das *Setup* beschreibt die genaue Prozedur der Simulation, die auf das Modell angewendet werden soll. In der Regel schließt sich an die eigentliche Simulation eine *Postprozessierung* der Rohdaten an, um die tatsächlichen Simulationsergebnisse zu gewinnen. Setup und Postprozessierung zusammen umfassen die pragmatische Funktion. Die Instantiierung und die initialen Werte der Zustandsvariablen des dynamischen Systems sollten zu den *Randbedingungen* und dem *initialen Zustand* des biologischen Systems passen, unter denen die zu erklärenden Phänomene beobachtet wurden. Außerdem sollten das Setup und die Postprozessierung die *experimentellen Settings* und die *Auswertung* widerspiegeln, mit der die biologischen Beobachtungen gewonnen wurden.

Die funktionale Bedeutung wird durch folgende Fragen charakterisiert:

#### Facette F1: Intention (teleologische Funktion)

**Intrinsisch:** Was ist die *intendierte Verwendung* des Modells? Welche *Constraints* müssen bei der Modellverwendung eingehalten werden?

Die intendierte Verwendung legt den Typ der Simulation und die Art der Ergebnisse fest. Für bestimmte Ergebnisarten (z.B. Bifurkationsdiagramme) kann es notwendig sein, verschiedene Simulationen miteinander zu kombinieren. Solche Kombinationen gehören ebenfalls zur intendierten Verwendung. Durch die Constraints werden Bedingungen für das dynamische System definiert, die bei der Simulation eingehalten werden müssen. Dabei kann es sich um erlaubte Wertebereiche von Attributen ([Facette S2](#)) und um Beziehungen zwischen Attributen handeln (z.B. Verhältnisse und Erhaltungssätze). Die Constraints werden in Form von Gleichungen und Ungleichungen gegeben. Diese dürfen auch unbestimmte Konstanten beinhalten, z.B. für die Aussage, dass sich die Summe mehrerer Attribute nicht ändert:  $x_1 + \dots + x_n = \text{const.}$  (in der in der Physik üblichen Notation).

**Extrinsisch:** Welche *biologischen Fragen* sollen durch die Verwendung des Modells beantwortet werden? Auf welchen *Annahmen* basieren die Constraints?

Bio-Modelle werden in Simulationen verwendet, um biologische Fragen zu beantworten. Damit sind hier aber nicht allgemeine biologische Fragen gemeint, sondern nur solche Fragen, die sich explizit an das Modell richten. Dabei soll durch die Simulation des Modells entweder beobachtetes Verhalten erklärt oder mögliches Verhalten des biologischen Systems ([Facette S1](#)) vorhergesagt werden. Die Erklärung umfasst dabei auch den Einfluss bestimmter Rand- und Anfangsbedingungen auf das Verhalten. Die Annahmen definieren einen Rahmen für die Beantwortung der biologischen Fragen.

### Facette F2: Instantiierung (konditionale Funktion)

**Intrinsisch:** Welche Modell-*Instantiierung* wird für die Simulation verwendet? Welche *Initialisierung* stellt den Ausgangszustand des dynamischen Systems bereit?

Ein Bio-Modell enthält formale Parameter und Attributvariablen. Für die Simulation muss das Modell instantiiert werden, d.h. alle formalen Parameter müssen aktuelle Werte erhalten. Außerdem müssen die Attributvariablen initiale Werte zugewiesen bekommen. Im Fall einer Kombination von Simulationen muss die Instantiierung und Initialisierung eventuell für jede Simulation erneut erfolgen, zumindest teilweise. Wird beispielsweise ein Bifurkationsdiagramm bezüglich eines Parameters erstellt, unterscheiden sich alle Modell-Instanzen genau in diesem Parameter. Die numerischen Werte für die aktuellen Parameter und die Attributvariablen sind so zu wählen, dass sie eine sinnvolle extrinsische Interpretation besitzen und die bei dieser Interpretation verwendeten Maßeinheiten miteinander verträglich sind (siehe [Abschnitt 2.1.3](#)).

**Extrinsisch:** Welche *Randbedingungen* entsprechen der Modell-Instantiierung? Welcher *initiale Zustand* des biologischen Systems entspricht den initialen Werten der Modell-Attribute?

Die Fragen, die mit Hilfe des Modells beantwortet werden sollen ([Facette F1](#)) beziehen sich auf bestimmte Randbedingungen und einen festgelegten Zustand, unter denen das biologische System experimentell beobachtet wird. Die Randbedingungen betreffen dabei die Umwelt des biologischen Systems wie Temperatur, Licht, Stoffkonzentrationen (insbesondere pH-Wert, Nährstoffe, Signalmoleküle) und räumliche Verhältnisse. Die kinetischen Daten, die als Grundlage für aktuelle Parameter dienen (siehe oben) hängen entscheidend von den Randbedingungen ab. Oftmals beziehen sich die biologischen Fragen auf plausible Bereiche von Randbedingungen und initialen Zuständen.

### Facette F3: Setup (pragmatische Funktion)

**Intrinsisch:** Welches *Setup* wird für die Simulation benutzt? Welche *Postprozessierung* der Rohdaten erzeugt die gewünschten Simulationsresultate?

Für die Verwendung eines Bio-Modells in einer Simulation muss im Setup die genaue Prozedur beschrieben werden, die auf das instantiierte und initialisierte Modell angewendet werden soll. Die Simulation liefert Rohdaten (siehe [Facette B1](#)). Um die finalen Simulationsresultate (vgl. [Facette B1](#)) zu erhalten, müssen diese Rohdaten eine Postprozessierung

durchlaufen, welche ebenfalls genau beschrieben werden muss. Im Falle einer Kombination von Simulationen ist zusätzlich detailliert anzugeben, wie die einzelnen Simulationen abgearbeitet und wie die einzelnen Ergebnisse zum Endergebnis kombiniert werden sollen.

**Extrinsisch:** Welches *experimentelle Setting* entspricht dem Setup der Simulation? Welche *Auswertung* transformiert die biologischen Messwerte in die experimentellen Messergebnisse?

Die Beantwortung biologischer Fragen (vgl. [Facette F1](#)) bezieht sich auf experimentelle biologische Beobachtungen. Entsprechend dem Setup der Simulation müssen die experimentellen Settings beschrieben werden, unter denen das biologische System beobachtet wurde bzw. unter welchen eine Überprüfung von vorhergesagtem Verhalten des biologischen Systems erfolgen soll. Dabei muss ebenfalls dargelegt werden, wie die Auswertung der biologischen Messwerte (vgl. [Facette B1](#)) zu den experimentellen Messergebnissen (vgl. [Facette B2](#)) führt.

### Beispiel 2.5 (Funktion von Modell 1 – Zeitreihe)

In diesem Beispiel soll gezeigt werden, wie das **Modell 1** verwendet wird, um eine spezielle Zeitreihe zu erzeugen. Diese Zeitreihe findet sich als Fig. 3a in [TYSON 1991](#) und wurde für [Abbildung 2.5](#) (linke Seite) reproduziert.

Die *intendierte Verwendung* ([Facette F1](#)) von **Modell 1** ist die Erzeugung einer Zeitreihe, genauer der zeitliche Verlauf der Werte von  $[M]/[CT]$  und  $[YT]/[CT]$ , wobei  $[YT] =_{df} [Y] + [YP] + [pM] + [M]$ . Für die Simulation legt [TYSON 1991](#) als *Constraints* ([Facette F1](#)) fest:

$$[CT] = \text{const.} \quad (2.4)$$

$$[aa] = \text{const.} \quad (2.5)$$

Die *Instantiierung* ([Facette F2](#)) des Modells für die Erzeugung der Zeitreihe besteht in der Vergabe von Werten für die formalen Parameter. In [Tabelle 1.2](#) sind die aktuellen Parameter für die Zeitreihe aufgelistet, wie sie in [TYSON 1991](#) zu finden sind. In [TYSON 1991](#) ist keine *Initialisierung* ([Facette F2](#)) für die Zeitreihe angegeben. Die initialen Werte für die Attribute in [Tabelle 1.2](#) sind geschätzte Werte aus Fig. 3a in [TYSON 1991](#) in Übereinstimmung mit dem Eintrag von **Modell 1** in der [BioModels Database](#). Entsprechend der Maßeinheiten für die kinetischen Konstanten in [Tabelle 1.2](#) ( $\text{min}^{-1}$ ) ist die Maßeinheit der Simulationszeit min (vgl. [Abschnitt 2.1.3](#)). Das *Setup* ([Facette F3](#)) zur Erzeugung der Zeitreihe besteht in der numerischen Integration des instantiierten und initialisierten Differentialgleichungssystems für  $t = 0 \text{ min}$  bis  $t = 100 \text{ min}$  mit einer Schrittweite von  $0.001 \text{ min}$ . Dafür muss ein geeigneter Integrationsalgorithmus gewählt werden. In [TYSON 1991](#) wird ein lineares Mehrschrittverfahren verwendet (Adams-Moulton-Algorithmus vierter Ordnung). Für die in [Abbildung 2.5](#) gezeigte Reproduktion der Zeitreihe wurde in [COPASI](#) der Algorithmus [LSODA](#) (*Livermore Solver for Ordinary Differential Equations with Automatic Method Selection*, [HINDMARSH 1983](#)) eingesetzt. Die *Postprozessierung* ([Facette F3](#)) besteht in diesem Beispiel aus der Berechnung der Werte von  $[M]/[CT]$  und  $[YT]/[CT]$  für jeden simulierten Zeitpunkt  $t$ .

[TYSON 1991](#) möchte mit Hilfe dieser Simulation ein bestimmtes biologisches Phänomen erklären. Die entsprechende *biologische Frage* ([Facette F1](#)) wird sehr konkret angegeben:

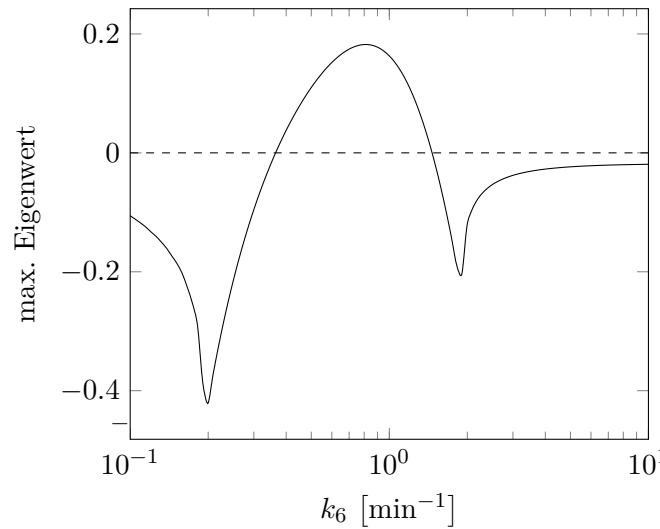


Abbildung 2.4.: Parameterscan bezüglich  $k_6$  des maximalen Eigenwertes der Jacobi-Matrix von Modell 1. Alle anderen Einstellungen wie in [Tabelle 1.2](#). Der Parameterscan wurde mit [COPASI](#) durchgeführt für  $k_6 = (0.1, 10) \text{ min}^{-1}$  (logarithmisch in 100 Schritten).

„Can the same model also account [...], for rapid cycles of DNA synthesis and cell division (without cell growth) during the embryonic cell cycle, [...].“  
([TYSON 1991](#), S. 7329)

Die *Annahme* ([Facette F1](#)), dass [CDK1](#) in wachsenden Zellen kontinuierlich in ausreichender Menge produziert wird ([TYSON 1991](#), S. 7329), stützt das oben genannte Constraint  $[CT] = \text{const.}$  Die Basis für das explizit benannte Constraint  $[aa] = \text{const.}$  ([TYSON 1991](#), S. 7329) bildet die implizite Annahme, dass Aminosäuren ( $aa$ ) in Zellen stets ausreichend vorhanden sind. Die *Randbedingungen* ([Facette F2](#)) und der *initiale Zustand* ([Facette F2](#)) entsprechen den Bedingungen, wie sie in frühembryonalen Zellen vorgefunden werden (vgl. [TYSON 1991](#), S. 7328). [TYSON 1991](#) (S. 7329) gibt an, dass für diese Situation keine kinetischen Daten existieren. In [TYSON 1991](#) werden keine weiteren Angaben zu den biologischen Beobachtungen gemacht, die erklärt werden sollen. Es fehlen also neben genauen Angaben zu Randbedingungen und zum initialen Zustand eine Beschreibung der extrinsischen pragmatischen Funktion ([Facette F3](#)) (*experimentelles Setting und Auswertung der Messwerte*). ■

### Beispiel 2.6 (Funktion von Modell 1 – Parameterscan)

In [Abbildung 2.4](#) ist ein Parameterscan von Modell 1 bezüglich  $k_6$  dargestellt, wobei alle anderen Werte wie in [Tabelle 1.2](#) gewählt sind. Dabei wird der Standardwert  $k_4 = 180 \text{ min}^{-1}$  (vgl. [Tabelle 1.2](#)) verwendet, der Scan entspricht also einem horizontalen Schnitt bei Punkt 3 in [Abbildung 1.4](#). Als abhängige Größe ist der maximale Eigenwert der Jacobi-Matrix von Modell 1 dargestellt. An den beiden Stellen, an denen der maximale Eigenwert 0 ist, tritt jeweils eine Bifurkation auf. Für eine ausführliche Analyse und Diskussion der entsprechenden Bifurkationen von Modell 2 siehe [Beispiel 5.1](#) und [Beispiel 5.2](#). ■



### 2.2.3. Verhalten eines Bio-Modells

Bio-Modelle werden in Simulationen verwendet, um ihr zeitliches Verhalten zu untersuchen. Die Simulationen produzieren *Rohdaten*. Aus den Rohdaten werden durch die Postprozessierung ([Facette F3](#)) die *Simulationsresultate* gewonnen. Die *Ausgabe* der Simulationsresultate erfolgt in Diagramm-Form oder als Report. Zusätzlich werden relevante *Charakteristiken* der Simulationsresultate ermittelt. Eine *Dynamik* ist eine qualitative Beschreibung der Ausgabe einer Simulation. Die *Diversifikation* stellt den Zusammenhang zwischen verschiedenen Dynamiken unter geänderten Bedingungen (Instantiierung oder Initialisierung) dar.

Bei einem Performanz-Modell korrespondiert die Dynamik, die das Modell in einer geeigneten Simulation zeigt, mit dem zu erklärenden *biologischen Phänomen*. Die Diversifikation der Dynamiken sollte die beobachtete *Variabilität* der biologischen Phänomene widerspiegeln. Biologische Phänomene sind dabei qualitative Beschreibungen entsprechender experimenteller *Messergebnisse*, welche in einer geeigneten *Darstellung* (als Messdiagramm oder als Messreihe) ausgegeben werden. Die Messergebnisse werden aus den *Messwerten* berechnet (siehe [Facette F3](#)), die im biologischen Experiment gewonnen werden und sind durch relevante *Merkmale* gekennzeichnet.

#### Facette B1: Daten

**Intrinsisch:** Welche *Rohdaten* werden von der Simulation erzeugt? Welche *Simulationsresultate* werden von der Postprozessierung aus den Rohdaten gewonnen?

Durch eine Simulation werden numerische Rohdaten erzeugt. Diese liegen als Folge von Werten in einem bestimmten Dateiformat vor und sind durch einen Index geordnet. Der Index kann z.B. die Simulationszeit  $t$  sein oder bei kombinierten Simulationen entsprechende Parameterwerte oder initiale Werte. Die Postprozessierung ([Facette F3](#)) transformiert die Rohdaten der Simulation in die finalen Simulationsresultate. Bei den Simulationsresultaten kann es sich um einzelne, spezifische Werte (z.B. Lage eines stationären Zustandes) und um Datenreihen (z.B. normierte Rohdaten) handeln. Für eine extrinsische Interpretation der Rohdaten müssen geeignete physikalische Maßeinheiten angegeben werden (vgl. [Abschnitt 2.1.3](#)).

**Extrinsisch:** Welche biologischen *Messwerte* aus zugehörigen Experimenten korrespondieren mit den Rohdaten aus der Simulation? Welche *Messergebnisse* korrespondieren mit den finalen Simulationsresultaten?

Biologische Experimente erzeugen Folgen von Messwerten. Diese sind ebenso indiziert wie die Rohdaten aus Simulationen. Die Auswertung ([Facette F3](#)) transformiert die experimentellen Messwerte in die Messergebnisse. Die Messergebnisse sind von der gleichen Art wie die finalen Daten der Simulation (siehe oben). Messergebnissen sind physikalische Maßeinheiten zugeordnet.

#### Facette B2: Ausgabe

**Intrinsisch:** Welche *Ausgabe* präsentiert die Simulationsresultate? Welche *Charakteristiken* der Simulationsresultate werden angegeben?

Häufig stellen die Simulationsresultate funktionale Abhängigkeiten zwischen bestimmten Größen dar, z.B. Zeitverläufe, Phasenportraits und Bifurkationsdiagramme. Diese werden in der Regel in geeigneter Form grafisch dargestellt. Die Beschreibung der grafischen Darstellung umfasst neben der Angabe, welche Daten dargestellt sind, den Typ (z.B. 2D-Diagramm, 3D-Diagramm), die Bedeutung und die Skalierung der Koordinatenachsen sowie die Darstellungsart (z.B. Punkte, Linien, Kurven). Neben der grafischen Darstellung können die Resultate der Simulation auch als Report ausgegeben werden. Dieser enthält neben ausgewählten Simulationsresultaten relevante Charakteristiken, wie z.B. maximale und minimale Werte, Mittelwerte, Perioden und Ljapunow-Exponenten. Ein Report sollte zudem eine genaue Beschreibung der Entstehung und Bedeutung der angegebenen Daten beinhalten.

**Extrinsisch:** Welche *Darstellung* der Messergebnisse korrespondiert mit der Ausgabe der Simulationsresultate? Welche *Merkmale* der Messergebnisse korrespondieren mit den Charakteristiken der Simulationsresultate?

Die Messergebnisse werden – analog zu den Simulationsresultaten – geeignet grafisch und als Report dargestellt und mit relevanten Merkmalen versehen.

### Facette B3: Dynamik

**Intrinsisch:** Welche *Dynamiken* zeigt das Modell in den Simulationen? Welche *Diversifikation* weisen die Dynamiken des Modells auf?

Eine Dynamik ist eine qualitative Beschreibung der Resultate einer Simulation. Die qualitative Beschreibung umfasst die Art des gezeigten Verhaltens (z.B. stationäres, periodisches und chaotisches Verhalten) und zusätzliche Bestimmungen (z.B. schnell/langsam, hoch/niedrig und steil/flach), welche in der Regel relativer Art sind, sich also auf andere Dynamiken beziehen. Die Diversifikation beschreibt Übergänge zwischen verschiedenen Dynamiken, wenn die Instantiierung oder die Initialisierung des Modells geändert wird. Im ersten Fall handelt es sich um eine Bifurkation, der zweite Fall bestimmt die Stabilität von Dynamiken. Die Beschreibung der Diversifikation beinhaltet die verschiedenen Dynamiken des Modells sowie die Typen und Bedingungen der Übergänge zwischen den Dynamiken.

**Extrinsisch:** Welche *biologischen Phänomene* korrespondieren mit den Dynamiken, die sich in der Simulation des Modells gezeigt haben? Welche *Variabilität* in den biologischen Phänomenen korrespondieren mit der Diversifikation der Dynamiken?

Die Beschreibung biologischer Phänomene auf der Grundlage von experimentellen Ergebnisse erfolgt in derselben Weise wie die qualitative Beschreibung der Simulationsergebnisse als Dynamik. Die Variabilität beschreibt qualitative Änderungen im Verhalten des biologischen Systems analog zur Diversifikation der Dynamiken des Modells.

### Beispiel 2.7 (Verhalten von Modell 1)

Die Simulation, deren Durchführung in [Beispiel 2.5](#) beschrieben ist, liefert für jedes der Attribute eine Reihe von Zahlenwerten als *Rohdaten* ([Facette B1](#)), die mit den Zeitschritten der Simulation indiziert sind. Durch einfache Berechnung werden daraus die *Simulationsresultate* ([Facette B1](#))  $[M]/[CT]$  und  $[YT]/[CT]$  gewonnen. Für die extrinsische Interpretation der Rohdaten und Simulationsresultate werden folgende physikalische Maßeinheiten gewählt:



Zeit (min)	$[M]/[CT]$	$[YT]/[CT]$
0	$0.000000 \cdot 10^0$	$2.500000 \cdot 10^{-1}$
0.1	$4.292610 \cdot 10^{-4}$	$2.515000 \cdot 10^{-1}$
0.2	$8.209520 \cdot 10^{-4}$	$2.529970 \cdot 10^{-1}$
0.3	$1.179810 \cdot 10^{-3}$	$2.544890 \cdot 10^{-1}$
0.4	$1.509820 \cdot 10^{-3}$	$2.559750 \cdot 10^{-1}$
0.5	$1.814390 \cdot 10^{-3}$	$2.574530 \cdot 10^{-1}$
0.6	$2.096430 \cdot 10^{-3}$	$2.589220 \cdot 10^{-1}$
0.7	$2.358440 \cdot 10^{-3}$	$2.603810 \cdot 10^{-1}$
0.8	$2.602590 \cdot 10^{-3}$	$2.618280 \cdot 10^{-1}$
0.9	$2.830770 \cdot 10^{-3}$	$2.632630 \cdot 10^{-1}$
1	$3.044620 \cdot 10^{-3}$	$2.646850 \cdot 10^{-1}$
...		
100	$3.396010 \cdot 10^{-3}$	$1.917270 \cdot 10^{-1}$

Tabelle 2.2.: Simulationsresultate für eine Zeitreihe von Modell 1 entsprechend Beispiel 2.5 zu ausgewählten Zeitpunkten (Spalte 1). Parameter und initiale Konzentrationen wurden wie in Tabelle 1.2 gewählt. Für Details siehe Beispiel 2.7.

Die Simulationszeit hat die Maßeinheit min, die Rohdaten sind in  $\text{mol} \cdot \text{l}^{-1}$  gegeben und die Simulationsresultate haben die Maßeinheit 1. Die Zahlenwerte werden als Gleitkommazahlen mit sechs signifikanten Dezimalstellen ausgegeben. Die Simulationsresultate der ersten zehn Zeitschritte sowie zu Beginn und am Ende der Simulation sind in Tabelle 2.2 aufgelistet. Die Ausgabe (Facette B2) der Simulationsresultate erfolgt als Graphen  $\{([M]/[CT], t)\}$  und  $\{([YT]/[CT], t)\}$ , welche entsprechend grafisch dargestellt werden können (Abbildung 2.5, links). Relevante Charakteristiken der Simulationsresultate sind die Periodenlänge von etwa 35 min (Tyson 1991, vgl.) sowie die minimalen und maximalen Werte der Oszillation:  $[YT]/[CT]$  schwingt zwischen etwa 0 und 0.2,  $[M]/[CT]$  zwischen etwa 0.1 und 0.35. Die Dynamik (Facette B3) dieser Zeitreihe ist eine stabile und gleichförmige (d.h. harmonische) Oszillation, welche in der Theorie dynamischer Systeme auch als stabiler Grenzyklus bezeichnet wird (siehe Abschnitt 5.1.2).

Modell 1 kann bei Veränderung der Instantiierung oder Initialisierung neben dieser Dynamik (stabiler Grenzyklus) auch noch andere Dynamiken aufweisen. Im Folgenden werden beide Arten der Diversifikation (Facette B3) betrachtet. Wird Modell 1 geringfügig anders initialisiert, zeigt sich ein Verhalten, das sich asymptotisch dem Grenzyklus nähert. Diese Näherung ist so schnell, dass nur anfangs eine kurze Einschwingphase zu beobachten ist. Neben dem stabilen Grenzyklus und sich diesem asymptotisch nähernden Verhaltensweisen besitzt Modell 1 für dieselbe Instantiierung noch eine dritte Dynamik: einen instabilen Fixpunkt (vgl. Abbildung 5.1). Wird Modell 1 genau in diesem Fixpunkt initialisiert, verlässt das System diesen Zustand nicht mehr. Eine Änderung der Instantiierung kann stärkere Folgen haben: Wird der Parameter  $k_6$  etwa von  $1 \text{ min}^{-1}$  auf  $2 \text{ min}^{-1}$  geändert (in Abbildung 1.4 durch ein Kreuz bzw. ein Plus markiert), bewirkt dies eine drastische Änderung der Dynamik des Systems (vgl. linke und rechte Seite von Abbildung 2.5). Für  $k_6 = 2 \text{ min}^{-1}$  weist Modell 1 einen stabilen Fixpunkt auf, das Verhalten strebt auf einen

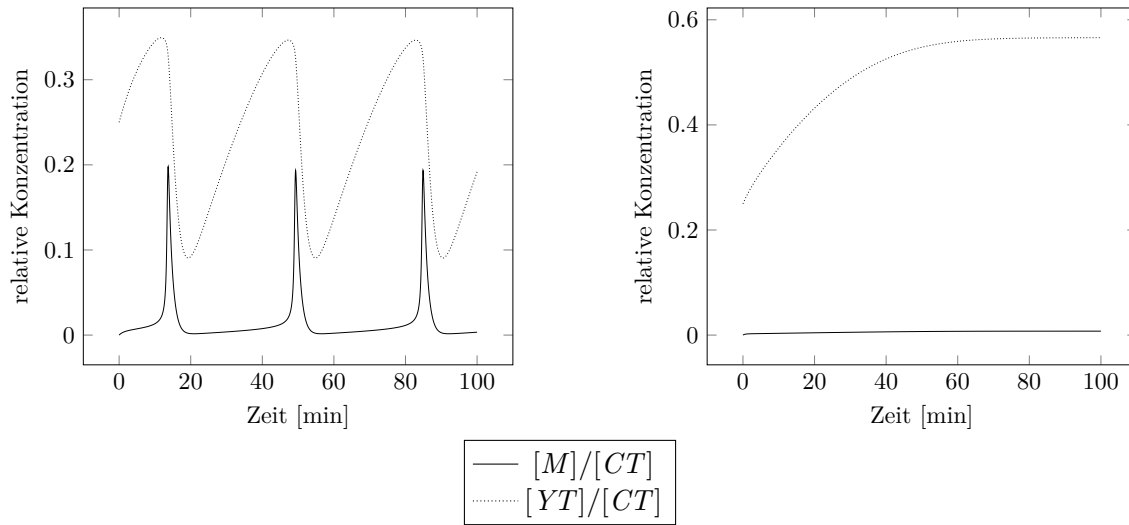


Abbildung 2.5.: Unterschiedliche Dynamiken von Modell 1 für  $k_6 = 1 \text{ min}^{-1}$  (links) und  $k_6 = 2 \text{ min}^{-1}$  (rechts). Alle anderen Werte wie in Tabelle 1.2. Für Details siehe Beispiel 2.5 und Beispiel 2.7.

stationären Zustand zu. Die Änderung von  $k_6$  führt zu einer Hopf-Bifurkation. Modell 1 weist eine weitere Hopf-Bifurkation bei Variation von  $k_6$  auf. Für Details siehe Beispiel 5.2. Ein Parameterscan (siehe Beispiel 2.6) kann für die Untersuchung der Diversifikation bei Variation eines oder mehrerer Parameter benutzt werden.

TYSON 1991 (S. 7328f) bezieht die Simulationsergebnisse auf unterschiedliche biologische Experimente. So wird z.B. die spezifische Charakteristik der (intrinsischen) Periodenlänge von etwa 35 min (siehe oben) auf eine Untersuchung von NEWPORT ET AL. 1982 bezogen. Dort wird als korrespondierendes (extrinsisches) Merkmal (Facette B2) eine Länge des Zellzyklus von  $(35 \pm 2) \text{ min}$  angegeben (NEWPORT ET AL. 1982, S. 677). Die Messwerte (Facette B1) dieser Untersuchung stammen aus der Beobachtung der Teilung individueller Zellen in Videoaufnahmen. Die jeweilige Dauer von der Entstehung einer Zelle bis zu ihrer Teilung (Länge des Zellzyklus) wird gemessen. Die Messergebnisse (Facette B1) sind die entsprechenden Mittelwerte der n-ten Teilung seit Befruchtung. Die Darstellung (Facette B2) der experimentellen Ergebnisse erfolgt grafisch als Diagramm der gemittelten Zellzykluslängen (in min) gegen die Nummer der Teilung (NEWPORT ET AL. 1982, Figure 2(A), S. 677).

Die Oszillation von  $M$ , wie sie sich bei dieser Simulation ergibt, wird von TYSON 1991 mit dem biologischen Phänomen (Facette B3) der schnellen Teilungszyklen in frühen Embryonen assoziiert, wie es etwa in NEWPORT ET AL. 1982 beschrieben ist. Der Bezug zwischen oszillierendem  $M$  und Teilungszyklen bedarf allerdings noch eines Zwischenschrittes: die Steuerung des Zellzyklus durch MPF (extrinsische Interpretation von  $M$ ). Dieser Zusammenhang ist biologisches Lehrbuchwissen (siehe z.B. KOOLMAN ET AL. 2003, S. 394f). Auf die experimentelle Evidenz für diesen Zusammenhang soll hier nicht näher eingegangen werden. TYSON 1991 beschreibt die Variabilität (Facette B3) der biologischen Phänomene und bezieht diese auf die unterschiedlichen Dynamiken von Modell 1, wie in Abschnitt 1.2.3 dargelegt. ■

## 2.3. Vergleich mit anderen Konzeptualisierungen

Im vorangegangenen [Abschnitt 2.2](#) wurden die Wissensfacetten von Bio-Modellen detailliert vorgestellt. Eine solche Konzeptualisierung der Modellierung ist nicht völlig neu. Die bereits existierenden Konzeptualisierungen sind jedoch aus verschiedenen Gründen unzureichend als Grundlage für eine umfassende Computerunterstützung der Bio-Modellierung, wie sie in [Abschnitt 1.3](#) beschrieben wurde. Die Wissensfacetten von Bio-Modellen stellen die notwendige konzeptuelle Basis für computergestützte Modellierung und Simulation bereit, insbesondere aus folgenden Gründen:

- Die ausführliche und einheitliche Behandlung der Struktur, Funktion und des Verhaltens von Bio-Modellen durch die Wissensfacetten ist Voraussetzung für eine systematischen Formalisierung relevanter Wissensfragmente.
- Die Trennung der intrinsischen und extrinsischen Aspekte sowie die Angabe der Unterfacetten dient als Wegweiser bei der Identifizierung relevanter Wissensfragmente.
- Die Wissensfacetten weisen einen hohen Detaillierungsgrad auf.

Im Folgenden sollen existierende Konzeptualisierungen der Modellierung zu den Wissensfacetten in Beziehung gesetzt werden. Nachdem in [Abschnitt 2.3.1](#) historische Überlegungen zur Dualität von Modellen diskutiert werden, betrachtet [Abschnitt 2.3.2](#) andere Ansätze, die sich mit Struktur, Funktion und Verhalten dynamischer Modelle auseinandersetzen. Abschließend werden in [Abschnitt 2.3.3](#) Arbeiten vorgestellt, die sich mit der biologischen Bedeutung von Bio-Modellen beschäftigen. Dabei wird insbesondere der Bezug zum biologischen Funktionsbegriff hergestellt.

### 2.3.1. Duale Interpretation von Bio-Modellen

Die Einsicht, dass mathematische Modelle eine duale Interpretation besitzen, ist nicht neu. Sie findet sich bereits in der Wissensrepräsentation. So fordert etwa [SMITH 1982](#) in der „knowledge representation hypothesis“ von einer nützlichen formalen Repräsentation, dass sie sowohl eine externe semantische Bedeutung tragen, als auch eine formale Rolle im computergestützten Wissensverarbeitungsprozess spielen soll:

„Any mechanically embodied intelligent process will be comprised of structural ingredients that a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exhibits, and b) independent of such external semantical attribution, play a formal but causal and essential role in engendering the behaviour that manifests that knowledge.“  
([SMITH 1982](#), S. 15)

Auch die Auffassung informatischer Artefakte als Schnittstellen zwischen einer inneren und äußeren Umgebung bei [SIMON 1969](#) erinnert an die duale Interpretation von Bio-Modellen.

Der Zusammenhang zwischen der *Modeling Relation* von [ROSEN 1985](#) und der dualen Interpretation von Bio-Modellen wurde bereits besprochen (siehe [Abschnitt 2.1.1](#) und [Abschnitt 1.1.1](#)) und wird in folgendem Zitat deutlich:

„The first basic dualism inherent in language is that (1) it is a thing in itself and (2) permits, even requires, referents external to itself.“ (ROSEN 1991, S. 43)

ROSEN formuliert seine *Modeling Relation* als Kongruenz zwischen einem natürlichen System und einem formalen System, wobei natürliche Objekte (bei ROSEN 1985 „percepts“) und Beziehungen zwischen diesen (bei ROSEN 1985 „linkages“) durch entsprechende formale Entitäten und Relationen codiert werden. Schlussfolgerungen in dem formalen System können dann als Prädiktionen über das natürliche System decodiert werden. In ROSEN 1985 findet sich also eine Unterscheidung zwischen der intrinsischen und der extrinsischen Interpretation sowohl aus struktureller Sicht als auch beim Verhalten der Systeme. Dabei liegt der Fokus auf dem Wechselspiel zwischen formalem und natürlichem System und nicht auf den Details der Struktur und des Verhaltens der Systeme.

Die Unterscheidung der intrinsischen und der extrinsischen Bedeutung ist für eine systematische formale Spezifikation von Bio-Modellen sehr hilfreich.

### 2.3.2. Drei Betrachtungsweisen von Bio-Modellen

Eine ähnliche Unterscheidung wie die drei Betrachtungsweisen von Bio-Modellen findet sich in CHELLIAH ET AL. 2009:

1. Die Modellbeschreibung („model description“) in CHELLIAH ET AL. 2009 entspricht ungefähr der hier beschriebenen strukturellen Sicht.
2. Die Beschreibung der Simulationen („simulation description“) von CHELLIAH ET AL. 2009 umfasst funktionale Aspekte.
3. Die Beschreibung der Simulationsresultate („simulation results description“) bei CHELLIAH ET AL. 2009 beinhalten Informationen über das Modellverhalten.

Die hier vorgestellten Wissensfacetten von Bio-Modellen sind jedoch systematischer und beinhalten mehr Details. In CHELLIAH ET AL. 2009 findet sich allerdings eine gute Übersicht existierender Standards, formaler Sprachen und Ontologien für die Bio-Modellierung.

### Funktionale Modellierung

In der funktionalen Modellierung werden Struktur, Funktion und Verhalten von technischen Artefakten miteinander in Beziehung gesetzt. Diese Dreiteilung war Inspiration für die drei Betrachtungsweisen von Bio-Modellen (Abschnitt 2.2).

ERDEN ET AL. 2008 stellt unterschiedliche Ansätze vor, die Funktion eines Artefakts und deren Beziehung zu Struktur und Verhalten zu formalisieren. Dabei werden zwei Auffassungen von „Funktion“ unterschieden: Einerseits vermittelt Funktion zwischen Struktur und Verhalten indem die möglichen Verhaltensweisen bestimmt werden, die ein Modell zeigen kann. Diese möglichen Verhaltensweisen bezeichnet ERDEN ET AL. 2008 entsprechend als „structural behaviours“. GOEL ET AL. 2009 (S. 24) sieht dabei die Bedingungen für ein bestimmtes Verhalten als Teil der Spezifikation einer Funktion. Diese Auffassung von Funktion entspricht somit im Wesentlichen der pragmatischen Funktion (Facette F3) und der konditionalen Funktion (Facette F2). Andererseits bezieht sich Funktion auf die Intentionen des Modellierers, wodurch die möglichen Verhaltensweisen des Modells auf die

erwarteten („expected behaviours“, [ERDEN ET AL. 2008](#)) beschränkt werden. In diesem Sinne ist Funktion eine „bridge between human intention and physical behavior of artifacts“ ([UMEDA ET AL. 1995](#), S. 271). Diese Auffassung entspricht der teleologischen Funktion ([Facette F1](#)).

Die Unterscheidung in strukturelles und erwartetes Verhalten geht zurück auf [GERO 1990](#). Die teleologische Funktion ([Facette F1](#)) sichert, dass durch die Verwendung des Modells, wie sie durch die konditionale und pragmatische Funktion spezifiziert wird, nur solche Verhaltensweisen betrachtet werden, die für die biologische Fragestellung von Bedeutung sind. In diesem Sinne stellt das Verhalten eines Bio-Modells ([Abschnitt 2.2.3](#)) erwartetes Verhalten dar.

### Modeling and Simulation

Das „framework for modeling and simulation“ von [ZEIGLER ET AL. 2000](#) stellt eine systematische Konzeptualisierung von Modellen und deren Verwendung dar. Das *Framework* von [ZEIGLER ET AL. 2000](#) besteht aus vier Aspekten: das Zielsystem ([ZEIGLER ET AL.](#) nennt dies „source system“, weil es die Quelle experimenteller Beobachtungsdaten ist), der experimentelle Rahmen („experimental frame“), das Modell selbst und den Simulator (vgl. [Abbildung 2.6](#)). Das spezifische Wissen über jedes dieser Elemente kann auf einem geeigneten „system specification level“ beschrieben werden (für Details siehe [ZEIGLER ET AL. 2000](#)). Es gibt einige Beziehungen zwischen dem *Framework* von [ZEIGLER ET AL. 2000](#) und den hier vorgestellten Wissensfacetten:

- Die Ebene 3 („state transition level“) von [ZEIGLER ET AL. 2000](#) entspricht dem *Programm* ([Facette S3](#)).
- Die Ebene 4 („coupled component level“) von [ZEIGLER ET AL. 2000](#) beschreibt die *Entitäten* ([Facette S2](#)) und *Relationen* ([Facette S3](#)) des Modells.
- Beide Ebenen zusammen werden in [ZEIGLER ET AL. 2000](#) benutzt, um das Element „Modell“ zu spezifizieren, also die Modell-Struktur festzulegen.
- Mit dem experimentellen Rahmen („experimental frame“) formalisiert [ZEIGLER ET AL. 2000](#) die Bedingungen für experimentelle Beobachtungen des Zielsystems. Dies entspricht den extrinsischen funktionalen Facetten der *Instantiierung* ([Facette F2](#)) und des *Setup* ([Facette F3](#)).
- Dabei wird gefordert, dass der experimentelle Rahmen eine „operational formulation of the objectives that motivate a modeling and simulation project“ ([ZEIGLER ET AL. 2000](#), S. 27) darstellt. Somit wird mit dem experimentellen Rahmen auch die *Intention* ([Facette F1](#)) der Modellverwendung festgelegt.
- Das Zielsystem („source system“) wird, wie oben beschrieben, als Quelle von Beobachtungsdaten angesehen. Diese Beobachtungsdaten werden bei [ZEIGLER ET AL. 2000](#) auf der Ebene 1 („I/O behaviour level“) beschrieben, was den experimentellen *Messwerten* ([Facette B1](#)) entspricht.

Es zeigt sich also, dass es Parallelen zwischen der Konzeptualisierung von [ZEIGLER ET AL. 2000](#) und den Wissensfacetten gibt. Allerdings fokussiert sich [ZEIGLER ET AL. 2000](#) auf die

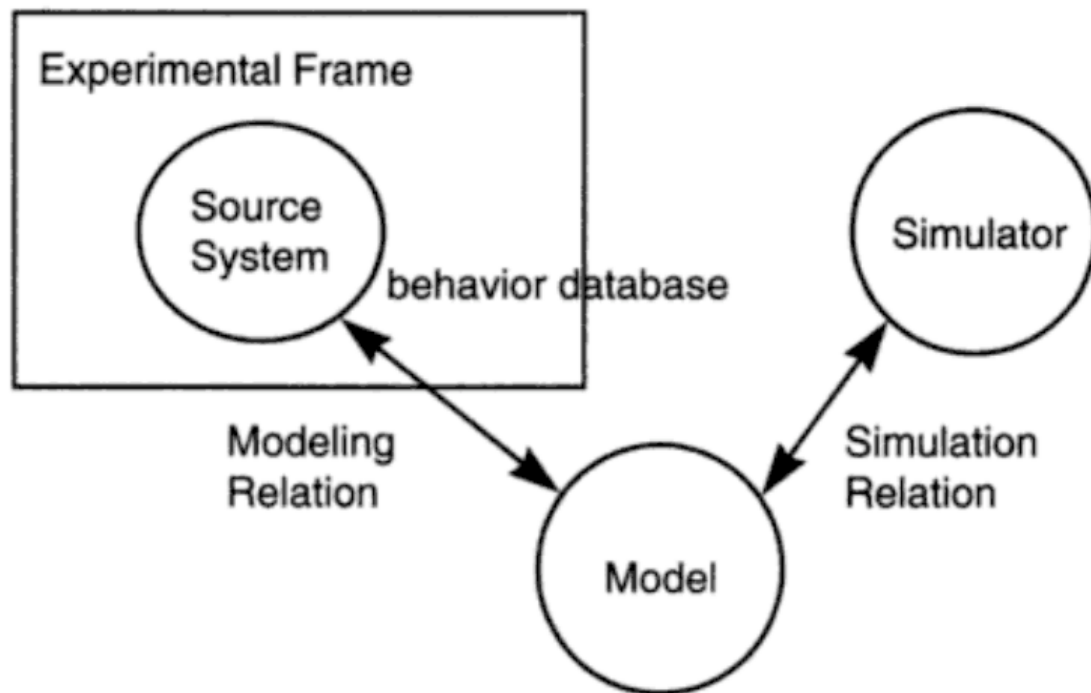


Abbildung 2.6.: Elemente der Konzeptualisierung bei ZEIGLER ET AL. Quelle: ZEIGLER ET AL. 2000 (Figure 1, S. 26).

mathematischen, d.h. intrinsischen, Aspekte der Bildung und Verwendung von Modellen. Dagegen werden hier Modelle als „integrators of knowledge“ (COURTOT ET AL. 2011), d.h. als das zentrale Vermittlungselement zwischen Computersimulationen und biologischer Wirklichkeit angesehen (vgl. Abbildung 2.1). Demzufolge fordern die Wissensfacetten auch eine detaillierte Darstellung der extrinsischen Interpretation eines Bio-Modells bezüglich Struktur, Funktion und Verhalten.

In KLIR 1985 wird das Wissen über untersuchte Systeme klassifiziert. Die dabei entwickelten „epistemological levels of systems“ ähneln ziemlich stark den „system specification levels“ von ZEIGLER ET AL. 2000. Tatsächlich nimmt ZEIGLER ET AL. 2000 die Beschreibungsebenen von KLIR 1985 als Ausgangspunkt für die eigene Konzeptualisierung und zeigt die Beziehungen zwischen beiden Ansätzen (ZEIGLER ET AL. 2000, S. 11ff).

### 2.3.3. Biologische Bedeutung von Bio-Modellen

Das Projekt SemSim (*Semantic Simulation*, NEAL ET AL. 2009) strebt eine Computerunterstützung für die Integration von Bio-Modellen an. Die Integration soll auf der Basis extrinsischer struktureller Informationen über die zu integrierenden Bio-Modelle erfolgen. Dafür wird die extrinsische Modellstruktur durch Annotationen mit Termen aus biologischen Ontologien festgelegt (GENNARI ET AL. 2008). Außerdem werden die *physikalischen Größen* (Facette S2), die durch Modellvariablen repräsentiert werden, mit Hilfe der eigens entwickelten OPB (*Ontology of Physics for Biology*, COOK ET AL. 2008) spezifiziert.

### Biologische Funktion

In der Biologie ist der Begriff der „Funktion“ allgegenwärtig. Es gibt starke Parallelen zwischen biologischer Funktion und der Funktion von Bio-Modellen, wie sie in [Abschnitt 2.2.2](#) beschrieben wird. Die den Wissensfacetten zugrunde liegende Idee, dass Funktion zwischen Struktur und Verhalten eines Bio-Modells vermittelt (siehe oben), findet sich auch in der Molekularbiologie, z.B. in folgendem Zitat:

„If one such behavior seems useful (to the organism), it becomes a candidate for explaining why the network itself was selected, i.e., it is seen as a potential purpose for the network.“ ([LANDER 2004](#), S. 0712)

In [KROHS ET AL. 2009](#) werden die Funktionsbegriffe der Biologie und Ingenieurwissenschaft miteinander verglichen und Unterschiede und Parallelen herausgearbeitet.

Es gibt Versuche, biologische Funktion formal zu erfassen. So werden die molekularen Funktionen biologischer Objekte von Ontologien wie [EcoCyc](#) ([KESELER ET AL. 2013](#)) und [GO](#) (*Gene Ontology*, [GENE ONTOLOGY CONSORTIUM 2001](#)) erfasst. In [BUREK ET AL. 2006](#) findet sich eine Konzeptualisierung biologischer Funktion, welche drei funktionale Aspekte formalisiert: die sogenannte Funktions-Struktur („function structure“), deren Realisierung („realization“) und die Relation „has-function“, welche einem biologischen Objekt eine Funktion zuordnet. Diese drei funktionalen Aspekte finden sich entsprechend als teleologische ([Facette F1](#)), pragmatische ([Facette F3](#)) bzw. konditionale ([Facette F2](#)) Funktion von Bio-Modellen wieder (vgl. [KNÜPFER ET AL. 2013a](#), S. 7).

Obwohl es starke Parallelen zwischen der Konzeptualisierung biologischer Funktion in [BUREK ET AL. 2006](#) und der hier vorgestellten Funktion von Bio-Modellen gibt, sind beide Ansätze nicht identisch. Das liegt insbesondere daran, dass Funktion in kategorial verschiedenen Bereichen betrachtet wird: [BUREK ET AL. 2006](#) betrachtet die Funktionen, die biologische Objekte in Lebensprozessen spielen. Die vorliegende Arbeit dagegen beschäftigt sich mit der Funktion informatischer Artefakte (Bio-Modelle) im wissenschaftlichen Prozess. An dieser Stelle sei darauf hingewiesen, dass die extrinsische Funktion eines Bio-Modells seine Verwendung zur Beantwortung biologischer Fragestellungen beschreibt und somit zu unterscheiden ist von den biologischen Funktionen, die im modellierten System zum Tragen kommen.

Der Begriff der biologischen Funktion war Inspiration für die Ausformulierung der funktionalen Wissensfacetten von Bio-Modellen. Darüber hinaus spielt die biologische Funktion in der vorliegenden Arbeit keine weitere Rolle.



### Zusammenfassung

In diesem Kapitel wurden die Wissensfacetten von Bio-Modellen vorgestellt (vgl. [Abbildung 2.7](#)): Dafür wurde zunächst untersucht, wie Bio-Modelle interpretiert werden ([Abschnitt 2.1](#)). Es wurde gezeigt, dass Bio-Modelle eine duale Interpretation besitzen: Einem Bio-Modell kann eine intrinsische formale Semantik und eine extrinsische biologische Bedeutung zugewiesen werden, wodurch eine Relation zwischen dem formalen und dem biologischen System hergestellt wird. Sowohl intrinsisch als auch extrinsisch kann ein Bio-Modell bezüglich seiner Struktur, seiner Funktion und seines Verhaltens betrachtet werden. In [Abschnitt 2.2](#) wurden die Wissensfacetten von Bio-Modellen hinsichtlich dieser Dreiteilung ausführlich untersucht. Abschließend wurde in [Abschnitt 2.3](#) das konzeptuelle Schema der Wissensfacette mit anderen Konzeptualisierungen der Modellierung in Beziehung gesetzt.

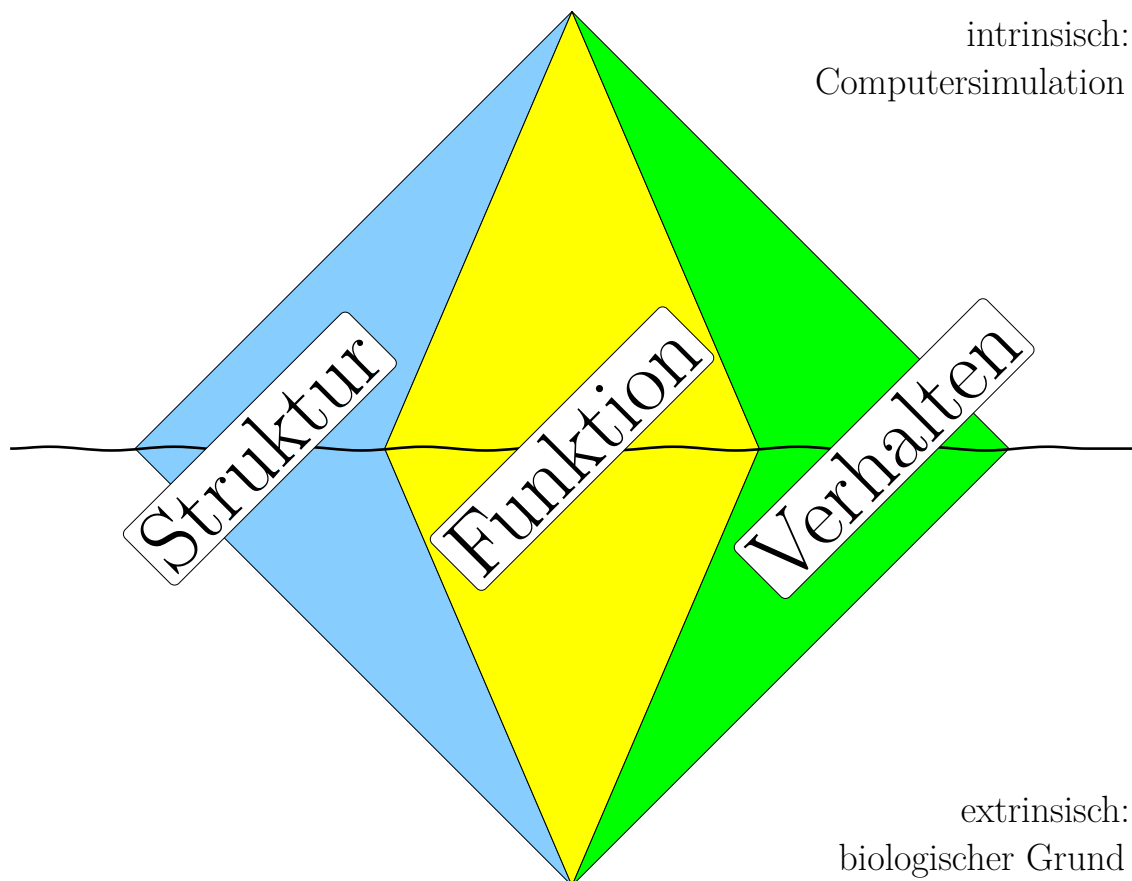


Abbildung 2.7.: Der „Wissensdiamant“ als kompakte Darstellung der Wissensfacetten von Bio-Modellen.

Zur Illustration der sechs identifizierten Wissensfacetten von Bio-Modellen dient der „Wissensdiamant“ in [Abbildung 2.7](#). Metaphorisch steht der Diamant zum einen für den Wert, den die Erschließung der Bedeutung von Bio-Modellen für Biologen darstellt und für die durch die Härte bedingten Schwierigkeiten, die diese Erschließung mit sich bringt. Zum

anderen wird die feste Einheit symbolisiert, die die Bedeutung eines Bio-Modells darstellt: Obwohl die Bedeutung verschiedene Facetten aufweist – der Diamant von verschiedenen Seiten betrachtet werden kann – bildet sie doch ein Ganzes. Aus Sicht einer computergestützten Behandlung von Bio-Modellen steckt der Diamant im „biologischen Grund“ (beachte auch die Anspielung auf das *Grounding*, siehe [Abschnitt 3.2.1](#)): Während die intrinsisch-mathematische Seite der Bedeutung eines Bio-Modells per definitionem formal ist, liegt die biologische Bedeutung eines Bio-Modells im allgemeinen als natürlichsprachlicher Text vor und ist somit einer computergestützten Behandlung nur sehr eingeschränkt zugänglich. In den Augen des Autors steht der Diamant auch für die konzeptionelle Schönheit der hier vorgelegten Analyse der Bedeutung von Bio-Modellen. Der Wissensdiamant ist eine komprimierte Version der grafischen Darstellung dynamischer Modellierung und Simulation in [Abbildung 1.8](#), welches am Ende von [Kapitel 1](#) präsentiert wurde. In beiden Abbildungen wurden dieselben Farben für die Struktur (Blau), die Funktion (Gelb) und das Verhalten (Grün) verwendet.

Die Wissensfacetten ermöglichen eine systematische Beschreibung eines Bio-Modells, seiner intendierten Verwendung in Simulationen und des dabei gezeigten Verhaltens. Die Struktur bestimmt die Komponenten, aus denen ein Bio-Modell besteht und setzt diese in Beziehung zum modellierten biologischen System. Dadurch wird das Modell zum Kompetenz-Modell, indem es mechanistische Erklärungen biologischer Phänomene ermöglicht. Die Funktion beschreibt die Konditionen und den Ablauf der intendierten Verwendung eines Bio-Modells in Simulationen zur Beantwortung biologischer Fragen bezüglich beobachteter zeitlicher Phänomene. Dabei wird eine Beziehung zu den biologischen Experimenten hergestellt, die zu den Beobachtungen führen. Das Verhalten stellt die Dynamik dar, die das Modell in den Simulationen zeigt und stellt eine Beziehung zu den biologischen Phänomenen und Beobachtungen her, die durch das Modell erklärt werden sollen. Dadurch wird das Modell zum Performanz-Modell bezüglich dieser Beobachtungen. Die systematische Beschreibung der Struktur, der Funktion und des Verhaltens eines Bio-Modells trägt wesentlich zum Austausch von Bio-Modellen zwischen Wissenschaftlern bei und erleichtert deren Verwendung und Wiederverwendung. Die Wissensfacetten stellen somit eine wichtige Voraussetzung für kollaborative Modellierung dar.

Für die Computer-Unterstützung der Modellierung ist es notwendig, das durch die Wissensfacetten systematisierte Wissen formal zu repräsentieren. Eine vollständige formale Repräsentation eines Bio-Modells muss seine Struktur, seine Funktion und sein Verhalten umfassen und die formale Semantik mit der biologischen Wirklichkeit in Beziehung setzen. Damit beschäftigt sich das [Kapitel 4](#). Zuvor werden im folgenden [Kapitel 3](#) die dafür erforderlichen methodischen Grundlagen der Wissensrepräsentation vermittelt.



## Teil II.

# Wissensrepräsentation für die Modellierung und Simulation

„As any other type of data in life sciences, models have to be stored, exchanged and re-used. This was made possible by the development of a series of standards, that, when used in conjunction, can cover the whole life-cycle of a model, including the specification of its structure and syntax, the simulations to be run, and the description of its behaviour and resulting numerical output.“

---

(CHELLIAH ET AL. 2009: *Data Integration and Semantic Enrichment of Systems Biology Models and Simulations*, S. 5)

Im zweiten Teil der Arbeit wird dargestellt, wie das für die Computerunterstützung der Modellierung und Simulation biologischer Systeme relevante Wissen formal repräsentiert werden kann. Mit den grundlegenden Methoden und Techniken der Wissensrepräsentation, die das notwendige Wissen einer maschinellen Verarbeitung zugänglich macht, beschäftigt sich das folgende [Kapitel 3](#). Damit wird eine methodische Basis für die folgenden Kapitel der Arbeit bereitgestellt. Anschließend wird entlang der Wissensfacetten von Bio-Modellen skizziert, wie relevante Wissenfragmente angemessen formalisiert werden können ([Kapitel 4](#)). Dabei wird – sofern möglich – auf existierende Standards, formale Sprachen und informatische Ontologien zurück gegriffen.



## 3. Wissensrepräsentation

„Knowledge representation is the application of logic and ontology to the task of constructing computable models for some domain.“

---

(SOWA 2000a: *Knowledge Representation: Logical, Philosophical and Computational Foundations*, S. XII)

In [Teil I](#) wurde beschrieben, wie Bio-Modelle in der Systembiologie verwendet werden, um dynamische Phänomene mechanistisch zu erklären ([Kapitel 1](#)) und welches Wissen dabei involviert ist ([Kapitel 2](#)). Es wurde gezeigt, dass für die Modellierung und Simulation biologischer Systeme Computerunterstützung benötigt wird. Die Realisierung einer Computerunterstützung erfordert es, relevante Wissensfragmente formal so zu repräsentieren, dass sie von Computern verarbeitet werden können. Die *Wissensrepräsentation* ist ein Teilbereich der Künstlichen Intelligenz, der sich mit der Darstellung und Verarbeitung von Wissen durch Computer beschäftigt. In diesem Kapitel werden grundlegende Begriffe und ausgewählte Verfahren der Wissensrepräsentation erläutert. Dies soll die methodischen Grundlagen für die formale Repräsentation der Wissensfacetten von Bio-Modellen ([Kapitel 4](#)) bereitstellen und helfen, existierende Ansätze zur formalen Beschreibung von Bio-Modellen einzuordnen und zu bewerten. In [Kapitel 5](#) werden spezielle Verfahren zur Wissensrepräsentation dynamischer Sachverhalte behandelt. Das Wissen über dynamische Vorgänge stellt zwar besondere Anforderungen an eine informatische Repräsentation, die entsprechenden Verfahren lassen sich jedoch problemlos in die in diesem Kapitel vorgestellte allgemeine Methodik der Wissensrepräsentation einordnen. Außerdem werden in diesem Kapitel die methodischen Voraussetzungen für die Entwicklung der Ontologie [TEDDY](#) (*TErminology for the Description of DYnamics*, [COURTOT ET AL. 2011](#)) geschaffen (siehe [Kapitel 6](#)).

Das Diktum von [SOWA 2000a](#) soll als leitendes Motiv für dieses Kapitel dienen. Die Logik dient dabei als grundlegendes Mittel zur formalen Darstellung von Wissen und stellt Schlussfolgerungsmechanismen zur Verfügung, die dieses Wissen nutzbar machen ([Abschnitt 3.1](#)). Informatische Ontologien erlauben die Festlegung der Bedeutung der logischen Ausdrücke ([Abschnitt 3.2](#)). In [Abschnitt 3.3](#) wird gezeigt, wie informatische Ontologien praktisch realisiert und sie somit in einem Wissensrepräsentationssystem („computable model“) verfügbar gemacht werden können. Der vorliegende Abschnitt stellt in semiotischer Sprache gesprochen syntaktische (Logik), semantische (Ontologie) und pragmatische (Realisierung informatischer Ontologien) Aspekte der Wissensrepräsentation vor.

### 3.1. Formale Darstellung von Wissen

In diesem Abschnitt wird zunächst skizziert, wie Wissen mit Hilfe logikbasierter formaler Sprachen repräsentiert wird ([Abschnitt 3.1.1](#)). Anschließend werden andere Formen

von rechnergestützter Wissensspeicherung und Verarbeitung vorgestellt ([Abschnitt 3.1.2](#)). Den Abschluss bilden methodische Überlegungen zur Repräsentation von Wissen ([Abschnitt 3.1.3](#)).

Die Wissensrepräsentation konstruiert formale Modelle der Wirklichkeit im Sinne von [Abschnitt 1.1](#). Sehr deutlich wird dies in folgendem Zitat:

„*A knowledge representation is a surrogate. Physical objects, events, and relationships, which cannot be stored directly in a computer, are represented by symbols that serve as *surrogates* for the external things. The symbols and the links between them form a model of the external system. By manipulating the internal surrogates, a computer program can simulate the external system or reason about it.*“ ([SOWA 2000a](#), S. 134)

Im Gegensatz zu Bio-Modellen bedient sich die Wissensrepräsentation jedoch der Logik als Ausdrucksmittel.

#### 3.1.1. Logikbasierte Wissensrepräsentation

Die Repräsentation von Wissen erfolgt mittels formaler Sprachen. Für die algorithmische Verarbeitung des repräsentierten Wissens müssen zusätzlich zur formalen Sprache geeignete Schlussfolgerungsmechanismen definiert werden. Eine formale Sprache mit entsprechenden Schlussfolgerungsmechanismen wird als *Kalkül* bezeichnet ([OWSNICKI-KLEWE ET AL. 2000](#), S. 161). Im Folgenden wird ein Kalkül zur Repräsentation und Verarbeitung von Wissen als *Repräsentationsformalismus* bezeichnet. Die Semiotik liefert wichtige Grundbegriffe zur Behandlung solcher Repräsentationsformalismen und zum Verhältnis zwischen Repräsentation und Repräsentiertem. Dieser Abschnitt untersucht die syntaktischen, semantischen und pragmatischen Aspekte von Repräsentationsformalismen. Zur Illustration dient uns dabei die Prädikatenlogik erster Stufe, hier kurz **FOL** (*First-Order Logic*) genannt.

Die semiotische Charakterisierung kann im Wesentlichen für jeden Repräsentationsformalismus durchgeführt werden, unabhängig von ihrer konkreten Syntax:

„[...] it is not a particular syntax for which logicians argue; it is the notion of a formal, declarative semantics and methods of inference that are important.“ ([VAN HARMELEN ET AL. 2008](#), S. 67)

Die **FOL** kann dabei als stellvertretend für alle algorithmisierbaren Repräsentationsformalismen gesehen werden. Alle existierenden Repräsentationsformalismen sind logikbasiert und können in **FOL** reformuliert werden (höherstufige Prädikatenlogiken sind nicht algorithmisierbar und sind deshalb keine geeigneten Repräsentationsformalismen):

„The same meanings can be expressed in a wide variety of syntactic forms. There is thus an a priori possibility that some already existing language may be best interpreted as another syntax for predicate (or even propositional) calculus.“ ([HAYES 1977](#), S. 561)

Mehr noch, Wissen muss durch Logik repräsentierbar sein, um rechnergestützt verarbeitet werden zu können:



„Perhaps there are some kinds of knowledge that cannot be expressed in logic. But if such knowledge exists, it cannot be represented or manipulated on any digital computer in any other notation.“ (SOWA 2000a, S. 12)

Das Erfüllbarkeitsproblem (siehe unten) ist für FOL-Formeln bekanntermaßen unentscheidbar. Aus diesem Grund werden häufig entscheidbare Fragmente der FOL oder diesen äquivalente formale Sprachen in der Wissensrepräsentation eingesetzt.

Die Logik kann also als Universalsprache für die Wissensrepräsentation aufgefasst werden. Diese Idee einer formalisierten Universalsprache und eines zugeordneten Schlussfolgerungsapparates findet sich bereits bei Leibniz:

„Leibniz’s program of a universal science (scientia universalis) for coordinating all human knowledge into a systematic whole comprises two parts: (1) a universal notation (characteristica universalis) by use of which any item of information whatever can be recorded in a natural and systematic way, and (2) a means of manipulating the knowledge thus recorded in a computational fashion, so as to reveal its logical interrelations and consequences (the calculus ratiocinator).“ (RESCHER 1954, S. 133)

Die Logik hat für die Wissensrepräsentation eine Bedeutung, die über die Bereitstellung einer Universalsprache hinausgeht:

„The reason logic is relevant to knowledge representation and reasoning is simply that, at least according to one view, logic *is* the study of entailment relations – languages, truth conditions, and rules of inference.“ (BRACHMAN ET AL. 2004, S. 11)

Die Logik stellt also den metasprachlichen, theoretischen Überbau der Wissensrepräsentation dar.

Obwohl also theoretisch die FOL als universelles Mittel dienen kann, wird in der Praxis die Wahl einer formalen Sprache für die Repräsentation von Wissen nach pragmatischen Kriterien erfolgen. So fordert etwa OWSNICKI-KLEWE ET AL. 2000 (S. 157f) von einer Repräsentationssprache Kommunikativität (das repräsentierte Wissen muss ausgetauscht werden können), Ausdrucksfähigkeit (die Sprache muss mächtig genug sein, um das relevante Wissen repräsentieren zu können) und Operationalisierbarkeit (die gestellten Aufgaben müssen mit dem repräsentierten Wissen algorithmisch gelöst werden können).

#### Syntax eines Repräsentationsformalismus

Ein Repräsentationsformalismus stellt eine formale Sprache zur Verfügung, mit deren Hilfe Wissen symbolisch repräsentiert werden kann. Die Festlegung einer formalen Sprache über einem Alphabet (endlicher Zeichenvorrat) erfolgt meist mit Hilfe einer Grammatik. Eine Grammatik ist eine Menge von Regeln, die angibt, wie aus den Zeichen wohlgeformte Ausdrücke der formalen Sprache (Zeichenketten, Wörter) gebildet werden können. Das Alphabet der FOL besteht aus Variablen-, Konstanten-, Funktions- und Prädikatensymbolen sowie aus Symbolen für die Junktoren und Quantoren. Daneben enthält das Alphabet noch Hilfszeichen, wie etwa Klammern. Mit Hilfe der Variablen-, Konstanten- und Funktionssymbole können induktiv Terme gebildet werden. Prädikatensymbole werden benutzt,

um atomare Ausdrücke über den Termen zu bilden. Aus atomaren Ausdrücken können mit den Junktoren und Quantoren induktiv zusammengesetzte Ausdrücke gebildet werden. Ausdrücke der FOL werden auch *Formeln* genannt. Für Details siehe z.B. SCHÖNING 2000.

#### Semantik eines Repräsentationsformalismus

Generell legt eine referentielle Semantik die Bedeutung von Ausdrücken in einem Repräsentationsformalismus mit Hilfe einer sogenannten *Interpretation* fest. Dabei werden Ausdrücke als Aussagen über einer gegebenen Menge von Dingen (*Individuen*) interpretiert. Diese Menge wird meist als *Domäne*, Universum oder Gegenstandsbereich bezeichnet. Die Gegenstände der Domäne sind Designate/Denotate für entsprechende elementare Symbole eines Ausdrucks. Regeln geben an, wie auf Grundlage der Bedeutung elementarer Symbole zusammengesetzte Ausdrücke interpretiert werden.

Die Symbole des Alphabets der FOL können, abgesehen von den Hilfszeichen, in zwei Gruppen eingeteilt werden: die logischen Symbole (Junktoren und Quantoren) und die nicht-logischen Symbole (Variablen-, Konstanten-, Funktions- und Prädikatensymbole). Während die Bedeutung der logischen Symbole von vornherein feststeht, müssen die nicht-logischen Symbole geeignet interpretiert werden, um FOL-Formeln eine Bedeutung zuweisen zu können. Die Bedeutung einer FOL-Formel ist ein Wahrheitswert (wahr oder falsch) und erfolgt mit Hilfe einer Interpretation entsprechend dem induktiven Aufbau der Formel: Jedem Term wird ein Element aus der Domäne zugeordnet und jedem Prädikat über entsprechende Individuen ein Wahrheitswert. Ausgehend von dieser Interpretation atomarer Ausdrücke werden die zusammengesetzten Ausdrücke induktiv interpretiert. Für Details siehe etwa SCHÖNING 2000.

Eine Interpretation, die eine FOL-Formel wahr macht, heißt *Modell*<sup>1</sup> dieser Formel. Eine Formel, die mindestens ein Modell besitzt, heißt *erfüllbar*, ansonsten *unerfüllbar*. Damit kann ein (semantischer) Folgerungsbegriff definiert werden: Eine Formel  $G$  ist *Folgerung* einer Formel  $F$  (geschrieben als  $F \models G$ ), falls jedes Modell von  $F$  auch Modell von  $G$  ist.

Die referentielle Semantik ist der Mechanismus, der es einem Repräsentationsformalismus ermöglicht, Wissen zu repräsentieren. Die Terme repräsentieren dabei Gegenstände (Individuen) des betrachteten Wirklichkeitsausschnitts (Domäne). Ausdrücke eines Repräsentationsformalismus repräsentieren Aussagen (Wissen) über die designierten Gegenstände. Im Sinne der Semiotik kann man auch davon sprechen, dass Ausdrücke (Zeichenfolgen) eines Repräsentationsformalismus Wissen *codieren*. Die Interpretation eines Ausdruckes ist dann eine Decodierung.

#### Pragmatik eines Repräsentationsformalismus

Der dritte Aspekt eines Repräsentationsformalismus ist seine Fähigkeit, das repräsentierte Wissen für Schlussfolgerungen zu nutzen. Hier ist die Unterscheidung zwischen der syntaktischen (symbolische) Ebene und der semantischen Ebene hilfreich. Der Schlussfolgerungsapparat eines Repräsentationsformalismus ermöglicht die rein mechanische *Ableitung* neuer Ausdrücke aus vorhandenen Ausdrücken durch Symbolmanipulationen. Ob diese Ableitungen angemessen sind, lässt sich nur auf der semantischen Ebene entscheiden. Auf

---

<sup>1</sup>In diesem Abschnitt ist „Modell“ im Sinne der Modelltheorie zu verstehen, vgl. Abschnitt 1.1.

der semantischen Ebene spricht man vom Schlussfolgern von Wissen auf der Basis von vorhandenem Wissen, ein entsprechender Folgerungsbegriff wurde oben eingeführt.

Für die **FOL** gibt es unterschiedliche Ableitungsregeln. Die wohl prominentesten sind der Modus Ponens und seine Verallgemeinerung, die Resolution. Die Resolution ist eine korrekte Ableitungsregel, d.h. aus wahren Formeln können nur wahre Formeln abgeleitet werden und zwar unabhängig von der Interpretation. Eine syntaktische Ableitung mit der Resolution kann also als semantische Folgerung (Deduktion) interpretiert werden. Für bestimmte Teilbereiche der **FOL** kann sogar gezeigt werden, dass es für jede mögliche Folgerung auch eine Ableitung durch Resolution gibt, die Resolution also *vollständig* ist. Für Details sei wieder auf die einschlägige Literatur verwiesen. Insbesondere stellt die effiziente Suche nach Ableitungen durch sogenannte Theorembeweiser einen umfangreichen Forschungsgegenstand dar, auf den hier nicht eingegangen werden soll.

### Repräsentiertes Wissen

Eine Formel repräsentiert einen einzelnen *Fakt* über einen Wirklichkeitsausschnitt. In der Wissensrepräsentation werden solche Fakten auch häufig als *Axiome* bezeichnet: Sie umfassen das explizit gegebene Wissen. Durch Ableitungen können neue Fakten generiert werden, die manchmal auch *Theoreme* genannt werden. Eine Sammlung von Fakten, die für die Lösung bestimmter Aufgaben eingesetzt werden kann, heißt *Wissensbasis*. Die Unterscheidung zwischen Axiomen und Theoremen spiegelt sich auch in den beiden elementaren Operationen vieler Wissensrepräsentationssysteme wieder: Mit 'TELL' werden der Wissensbasis neue Fakten mitgeteilt und mit 'ASK' Anfragen gestellt und damit gegebenenfalls Schlussfolgerungen veranlasst (vgl. [NEBEL 1990](#), S. 14).

#### 3.1.2. Wissen und Daten

Neben der klassischen, logischen Wissensrepräsentation gibt es noch andere Formen der formalen Darstellung von Wissen. Für diese Arbeit interessant sind dabei die Strukturierung von Wissensinhalten in spezialisierten formalen Sprachenschemata, die Speicherung von Wissen in Datenbanken, die verteilte Ablage und Verlinkung von Wissen im Internet und Metainformationen über die Repräsentation von Wissen.

### Formale Sprachenschemata

Formale Sprachenschemata erlauben die Repräsentation sehr spezialisierter Wissensinhalte in strukturierter Form. Das Schema gibt dabei die Struktur (Grammatik) der verwendeten formalen Sprache vor und legt fest, wie zusammengesetzte Ausdrücke der formalen Sprache interpretiert werden. Solche Sprachenschemata finden sich auf verschiedenen Abstraktionsstufen. So stellt z.B. **OWL** (*Web Ontology Language*, [HITZLER ET AL. 2012](#)) ein Schema zur Repräsentation von Ontologien dar (siehe [Abschnitt 3.3.2](#)). Zentrales Sprachelement von **OWL** ist `class`, mit welchem Konzepte definiert und beschrieben werden können. Eine konkreteres Sprachschema stellt **SBML** (*Systems Biology Markup Language*, [HUCKA ET AL. 2003](#)) dar, wobei die zentralen Elemente `species` und `reaction` der Repräsentation von chemischen Spezies und Reaktionen dienen (vgl. [Abschnitt 4.2.1](#)).

Sprachenschemata können Sprachhierarchien bilden. Dabei werden Sprachelemente des konkreteren Schemas als Inhalte geeigneter Elemente des abstrakteren Schemas reifiziert. So

kann z.B. das Sprachelement **species** von [SBML](#) als **class** in [OWL](#) definiert werden. Dabei geht jedoch die fest verankerte Interpretation des konkreteren Sprachelements verloren und muss geeignet in der abstrakteren Sprache festgelegt werden.

#### Datenbanken

Auch in Datenbanken wird Wissen, insbesondere Faktenwissen, gespeichert. Dabei gibt das Datenbankschema die Interpretation der Daten vor. Im Kontext dieser Arbeit spielen insbesondere die zahlreichen biologischen Datenbanken eine Rolle, die Wissen über chemische Verbindungen und Reaktionen enthalten. Diese Datenbanken können dem *Grounding* von Modellelementen dienen (vgl. [Abschnitt 3.2.1](#)).

#### Wissen im Internet

Das Internet bietet vielfältige Möglichkeiten, Wissen verteilt zu repräsentieren und miteinander in Beziehung zu setzen. An dieser Stelle sollen nur die wichtigsten Schlagwörter genannt werden. Insbesondere das *Semantic Web* hat es sich zur Aufgabe gemacht, Wissen formal zu repräsentieren. Im Ansatz „linked open data“ geht es um die Verknüpfung von frei zur Verfügung gestellten Daten. Eine andere Form, Wissensfragmente aufeinander zu beziehen, stellen sogenannte *Annotations* dar. Durch Annotationen werden zusätzliche, externe Informationen zugänglich gemacht.

Eine wichtige Sprache für die formale Repräsentation von Beziehungen zwischen Wissensfragmenten im *Semantic Web* ist [RDF](#) (*Resource Description Framework*, [MANOLA ET AL. 2004](#)). Oft werden auch Annotationen mit Hilfe von [RDF](#) formuliert. Sogenannte *Triplestores* stellen Sammlungen solcher in [RDF](#) formulierten Beziehungen dar. Mit [SPARQL](#) (*SPARQL Protocol And RDF Query Language*, [HARRIS ET AL. 2013](#)) können Anfragen an einen *Triplestore* formuliert werden.

Die Codierung formaler Sprachen (siehe oben) für den Austausch im Internet erfolgt zumeist in Form einer *Markup Language*, meist als Erweiterung von [XML](#) (*Extensible Markup Language*, [BRAY ET AL. 2006](#)). Das Sprachschema wird dabei über ein [XML](#)-Schema definiert. Damit wird zugleich auch das Datenformat festgelegt, mit dem die formale Sprache codiert wird.

Wichtig für die Referenzierung von Wissensfragmenten im Internet ist die Verfügbarkeit global eindeutiger Bezeichner. Als solche werden im *Semantic Web* häufig [URIs](#) (*Uniform Resource Identifier*) bzw. deren internationalisierte Version, die [IRIs](#) (*Internationalized Resource Identifier*) verwendet. Dienste wie <http://purl.org/> und [identifiers.org](http://identifiers.org) ([JUTY ET AL. 2012](#)) stellen [PURLs](#) (*Persistent Uniform Resource Locator*) für registrierte Internet-Ressourcen bereit und abstrahieren damit von den eigentlichen Internet-Adressen der referenzierten Daten, die sich im Laufe der Zeit ändern können. In ähnlicher Weise abstrahieren [URNs](#) (*Uniform Resource Name*), eine spezielle Form von [URIs](#), von den eigentlichen Internet-Adressen der referenzierten Daten.

#### Metainformationen

Für die Bewertung und Auswahl von repräsentiertem Wissen ist es hilfreich, wenn Informationen über dieses Wissen und über den Repräsentationsvorgang zugänglich sind. Solche Metadaten oder Metainformationen stellen selbst repräsentiertes Wissen dar. Typische

Metainformationen sind z.B. die Urheberschaft des repräsentierten Wissens (Autoren, Kuratoren), das Datum der Erstellung eines Wissensfragments sowie Informationen über die Versionsgeschichte und Zugriffsrechte. Metainformationen werden oft in [RDF](#) beschrieben und verweisen dann gegebenenfalls in Form von Annotationen auf anderes Wissen.

### 3.1.3. Knowledge Engineering

Dieser Abschnitt beschreibt das Vorgehen bei der Entwicklung einer Wissensbasis. Eine ausführliche Darstellung der verschiedenen Methodologien und Verfahren des Knowledge Engineerings ist im Rahmen dieser Arbeit weder möglich noch notwendig. Stattdessen sollen hier die wichtigsten Schritte des Entwurfs einer Wissensbasis kurz dargestellt werden und relevante Begriffe eingeführt werden.

Das Gebiet des Knowledge Engineerings kann treffend so charakterisiert werden:

„Knowledge engineering can therefore be defined as the branch of engineering that analyzes knowledge about some subject and transforms it to a computable form for some purpose.“ ([SOWA 2000a](#), S. 132)

In diesem Zitat werden schon die beiden Hauptphasen des Knowledge Engineerings benannt: Analyse und Formalisierung. Etwas detaillierter beschreibt [HAYES 1985b](#) (S. 31f) das Vorgehen, welches als „obvious methodology“ bezeichnet wird:

1. introspektive Befragung eines Experten,
2. Erstellung einer vorläufigen Formalisierung,
3. Benutzung der vorläufigen Formalisierung zur Problemlösung,
4. kritische Bewertung der Performanz der Problemlösung durch den Experten,
5. Verbesserung der Formalisierung und
6. Wiederholung der Benutzung, Bewertung und Verbesserung.

Die Methode von HAYES beschreibt das Wechselspiel zwischen Fachexperten und Wissensingenieuren, sagt aber noch nichts darüber, wie man von dem Wissen eines Wirklichkeitsausschnitts zu einer formalen Repräsentation kommt. Es gibt verschiedene Vorgehensweisen bei der Entwicklung einer Wissensbasis. Eine Übersicht der verschiedenen Ansätze findet sich z.B. in [PUPPE ET AL. 2000](#) (S. 603, S. 625ff). Folgende Phasen finden sich aber mehr oder weniger explizit in all diesen Ansätzen:

1. Wissensakquise: Erwerb und Sammlung relevanter Wissensfragmente,
2. Konzeptualisierung: ontologische Analyse des erworbenen Wissens,
3. Formalisierung: Spezifikation entsprechender formaler Sprachen und Ontologien und
4. Verwendung: Codierung von Wissen und Anfragen an die Wissensbasis.

Diese Entwicklungsschritte stellen gleichzeitig den Lebenszyklus einer Wissensbasis dar. Entdeckte Fehler oder geänderte Anforderungen können zur Revision der Wissensbasis führen. Die Auswirkungen einer geänderten Formalisierung auf bereits repräsentiertes Wissen ist ein nicht-triviales Problem (vgl. [NEBEL 1990](#)).

#### Konzeptuelles *Framework*

Das Vorgehen beim Knowledge Engineering ist stark inspiriert von der Softwaretechnik. Das Ergebnis der Konzeptualisierungsphase wird dort häufig als „Domainmodell“ oder „konzeptuelles Modell“ bezeichnet. Im Kontext dieser Arbeit sollen diese traditionellen Begriffe vermieden werden, um nicht durch weitere Bedeutungen von „Modell“ zu Missverständnissen zu führen. Stattdessen wird das Ergebnis der ontologischen Analyse als „konzeptuelles *Framework*“ bezeichnet werden (vgl. [KNÜPFER ET AL. 2013a](#)).

#### Wissensakquise und Wissensspeicherung

Es gibt verschiedene Wege, wie das Wissen gewonnen wird, das die Grundlage für die Konzeptualisierung bildet bzw. später die Wissensbasis füllt. Die Bandbreite reicht von Expertenbefragungen über Literaturrecherche bis zu Text-Mining und Data-Mining (vgl. [PUPPE ET AL. 2000](#), S. 606ff). Das Füllen von Wissensbasen und Datenbanken erfolgt entweder überwacht durch speziell geschulte Kuratoren oder durch beliebige Benutzer. Letztere Methode hat insbesondere mit dem Web 2.0 einen gewissen Aufschwung erfahren. Dem Vorteil des schnelleren Wissensgewinns steht der Nachteil eventuell fehlerhafter, ungeprüfter Daten gegenüber. In jedem Fall ist es für die Qualität des repräsentierten Wissens und für die Geschwindigkeit der Eingabe vorteilhaft, wenn die Prozesse der Wissensakquise und der Wissensspeicherung computerunterstützt ablaufen. Dabei erlaubt bereits repräsentiertes Wissen die Strukturierung neuen Wissens und die Prüfung auf Verträglichkeit (Konsistenz). Außerdem können Verfahren der Autovervollständigung die Geschwindigkeit der Eingabe entscheidend erhöhen.

#### Kollaborative Entwicklung formaler Sprachen

Formale Sprachen, die für den Austausch von Wissen innerhalb einer Fachgemeinschaft konzipiert werden, verlangen bei der Entwicklung ein gewisses Maß an Einigung („shared agreement“) über Form und Inhalt des repräsentierten Wissens. Eine solche Entwicklung geht gezwungenermaßen mit einer Standardisierung sowohl der verwendeten Terminologie als auch der Abgrenzung relevanter Wissensfragmente einher. Entsprechende Richtlinien für die Darstellung eines bestimmten Typs von Wissen werden oft als [MI-Standards](#) (minimale Informations-Standards) bezeichnet (vgl. [Abschnitt 4.1.2](#)). Ein [MI-Standard](#) gibt dabei vor, welche Wissensfragmente repräsentiert werden müssen und wie diese strukturiert sein sollen. Damit kann ein [MI-Standard](#) als konzeptuelles *Framework* aufgefasst werden und als Grundlage für die Entwicklung entsprechender formaler Sprachen dienen.

## 3.2. Ontologien zum *Grounding* von Symbolen

In diesem Abschnitt soll untersucht werden, wie informatische Ontologien für das *Grounding* von Symbolen benutzt werden. Dafür wird zunächst das *Symbol Grounding Problem* diskutiert ([Abschnitt 3.2.1](#)) und anschließend gezeigt, wie der ursprünglich philosophische Begriff „Ontologie“ in der Informatik Verwendung gefunden hat ([Abschnitt 3.2.2](#)). Damit informatische Ontologien ihrer genuinen Aufgabe in der Wissensrepräsentation – Bereitstellung elementarer Konzepte zum *Grounding* von Symbolen – gerecht werden können, müssen



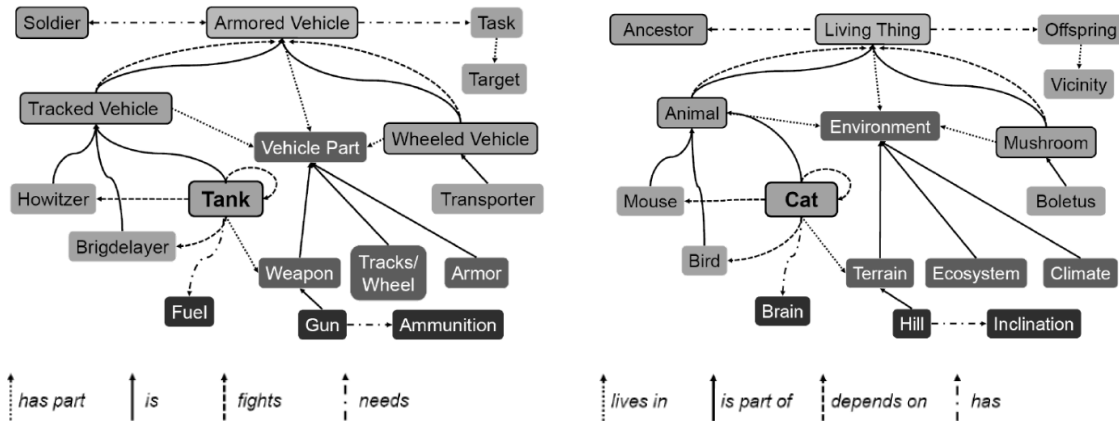


Abbildung 3.1.: Illustration des *Symbol Grounding Problems* (aus [HOFMANN 2013](#), S. 73, Fig. 5 und Fig. 6): Trotz struktureller Identität zwischen der linken und der rechten Abbildung auf symbolischer Ebene unterscheidet sich Semantik der beiden Abbildungen erheblich.

sie zwei Welten miteinander verbinden: die Symbol-Welt und die repräsentierte Wirklichkeit. Im abschließenden [Abschnitt 3.2.3](#) wird gezeigt, dass Ontologien logisch präzise und ästhetisch prägnant spezifiziert sein müssen, um diese Verbindung zu gewährleisten.

Der Darstellung informatischer Ontologien in diesem Abschnitt liegt folgende These zugrunde: Ontologien sind *das* informatische Mittel zum *Grounding* von Symbolen. Ontologien werden verwendet, um Symbolen, die in natürlichsprachlichen Texten und in Datenbankeinträgen vorkommen, Bedeutungen zuzuordnen. Dies geschieht durch Annotationen der Symbole, d.h. durch entsprechende Verweise vom Auftreten eines Symbols zu entsprechenden Ontologie-Konzepten (vgl. [Abschnitt 3.1.2](#)).

### 3.2.1. Das *Symbol Grounding Problem*

In diesem Abschnitt wird diskutiert, ob und wie formale Systeme überhaupt Wissen verarbeiten können. Die Algorithmen, die auf Repräsentationsformalismen operieren, arbeiten auf Grundlage von symbolisch repräsentiertem Wissen, d.h. auf der syntaktischen Ebene. Sie haben dabei keinen Zugang zur Interpretation des repräsentierten Wissen, d.h. die semantische Ebene ist ihnen nicht zugänglich. Das sogenannte „*Symbol Grounding Problem*“ beschreibt die Schwierigkeit, einer symbolverarbeitenden Maschine die Bedeutungen ihrer Symbole zugänglich zu machen. Natürlich ist es möglich, die Bedeutung eines Symbols durch andere Symbole zu beschreiben. Das Problem verschiebt sich damit aber nun auf die Bedeutung der in dieser Beschreibung vorkommenden Symbole.

In [HOFMANN 2013](#) findet sich eine anschauliche Illustration des *Symbol Grounding Problems* ([Abbildung 3.1](#)). Aus (der symbolischen) Sicht einer formalen Repräsentation sind beide Repräsentationen bis auf Bezeichner der Knoten und Kanten (strukturell) identisch, obwohl sie völlig verschiedenes Wissen repräsentieren. Erst eine geeignete Interpretation dieser Bezeichner kann auf der semantischen Ebene diese unterschiedliche Bedeutung aufdecken.

Das *Symbol Grounding Problem* tauchte ursprünglich in der KI im Zusammenhang mit



der Frage auf, ob ein Computer „intelligent“ sein kann. Ausgehend von Searles Gedankenexperiment des chinesischen Zimmers (siehe etwa [ZIMMERLI ET AL. 1994](#), S. 21f), formulierte [HARNAD 1990](#) das *Symbol Grounding Problem* wie folgt:

„How can the semantic interpretation of a formal symbol system be made intrinsic to the system, rather than just parasitic on the meanings in our heads? How can the meanings of the meaningless symbol tokens, manipulated solely on the basis of their (arbitrary) shapes, be grounded in anything but other meaningless symbols?“ ([HARNAD 1990](#), Abstract)

Die philosophischen Diskussionen um die „künstliche Intelligenz“ sollen hier keine weitere Rolle spielen. Eine Sammlung der klassischen Texte zu diesem Thema findet sich in [ZIMMERLI ET AL. 1994](#).

Das *Symbol Grounding Problem* ist dabei kein ausschließliches Problem der KI. Es ist ein Problem jedweder Repräsentation. Auch die vollständige Beschreibung der Bedeutung eines deutschen Wortes unter alleiniger Zuhilfenahme anderer deutscher Wörter kann nicht gelingen. Die Bedeutung einiger Wörter muss immer außersprachlich festgelegt sein, um einen unendlichen Regress zu vermeiden. Zu Recht weist dementsprechend [SMITH 2008](#) darauf hin, dass die beschriebene Problematik des Bezuges von Symbolen zur repräsentierten Wirklichkeit (als Vertreter eines starken Realismus spricht SMITH von „Realität“) nicht spezifisch für die KI sind, sondern einfach in der Natur von Repräsentation liegt:

„Es scheint nun eine Grundannahme vieler Arbeiten auf dem Gebiet der Wissensrepräsentation zu sein, dass sich diesbezüglich etwas ändert, wenn Computer ins Spiel kommen und wissenschaftliche Lehrbücher durch elektronisch gespeicherte Daten ergänzt werden – als ob in Computern gespeicherte Ausdrücke aus irgendeinem Grund unfähig sind, sich auf dieselbe Weise wie Ausdrücke in gedruckten wissenschaftlichen Texten auf Entitäten in der Realität zu beziehen. [...] Diese Begründung zeigt aber, dass sie einfach die Natur der Repräsentation missverstehen.“ ([SMITH 2008](#), S. 44)

In dieser Sichtweise ist wohl auch die Deklarierung als „Problem“ unangemessen.

Neben der philosophischen Dimension des *Symbol Grounding Problems* hat es eine ganz praktische Bedeutung bei der computergestützten Repräsentation und Verarbeitung von Wissen:

„By itself, logic says nothing about anything, but the combination of logic with an ontology provides a language that can express relationships about the entities in the domain of interest.“ ([SOWA 2000a](#), S. 492)

Allgemein spielt die Bedeutung von Symbolen immer dann eine Rolle, wenn ein Übergang zwischen der symbolischen Ebene und der semantischen Ebene notwendig ist. Ganz konkret können die folgenden Aktionen der Wissensrepräsentation identifiziert werden, in welchen die Bedeutung wichtig ist:

1. Codierung des Wissens (Softwareentwicklung),
2. Interaktion mit dem Benutzer (Eingabe),

3. Verwendung des repräsentierten Wissens (Schlussfolgerungen) und
4. Decodierung der Ergebnisse (Ausgabe, Visualisierung).

Die Bedeutung der Symbole eines Repräsentationsformalismus spielt also zum einen an der Schnittstelle zwischen Computer und Benutzer eine Rolle. Zum anderen muss der Schlussfolgerungsapparat so gestaltet sein, dass Ableitungen auf der symbolischen Ebene sinnvollen Folgerungen auf der Wissensebene entsprechen (siehe [Abschnitt 3.1.1](#)).

Im Laufe der Diskussion des *Symbol Grounding Problems* wurden zahlreiche Vorschläge zu dessen Lösung gemacht. Bereits [HAYES 1985b](#) (S. 12f) weist auf drei Möglichkeiten hin, einem Symbol eine Bedeutung zuzuweisen: (1) Beschreibung der Bedeutung in einer Metasprache. Dies verschiebt das Problem jedoch nur, da die Symbole der Metasprache wiederum interpretiert werden müssen. (2) Sensorisch-motorische Bindung von Symbolen an die externe Umgebung (oft auch als „embodiment“ bezeichnet). (3) Festlegung der Symbolbedeutung mittels natürlicher Sprache durch ein Sprachverarbeitungssystem.

Die Lösungen, die auf der Verbindung von Symbolen mit Sensor-Eingaben basieren, verwenden oft konnektionistische Ansätze, um Sensordaten zu klassifizieren. Die zu einem Symbol gehörige Klasse von Sensordaten, d.h. seine Bedeutung, wird dabei erlernt. Die Bedeutung der Symbole wird also induktiv geschlossen. Bereits [HARNAD 1990](#) schlägt ein hybrides System vor, in welchem künstliche neuronale Netze verwendet werden, um elementare Symbole mit Sensordaten zu verbinden. Die elementaren Symbole bilden die Basis für die Festlegung der Bedeutung komplexer Symbole. [ZIEMKE 1999](#) gibt eine gute Übersicht über die unterschiedlichen Lösungsansätze für das *Symbol Grounding Problem*.

Konnektionistische Ansätze werden oft auch als „sub-symbolisch“ bezeichnet, da keine expliziten Symbole zur Repräsentation eingesetzt werden. [GÄRDENFORS 2004](#) schlägt einen Zwischenweg zwischen expliziter symbolischer Repräsentation und sub-symbolischer Repräsentation durch ein konnektionistisches System vor: In „conceptual spaces“ werden Konzepte als geometrische Orte in Merkmalsräumen festgelegt, d.h. als Teilmengen von Merkmalsräumen mit bestimmten topologischen Eigenschaften. Die so definierten Konzepte können zur Festlegung der Bedeutung von Symbolen verwendet werden. Somit werden Symbole der symbolischen Ebene auf der „konzeptuellen Ebene“ verankert. Die Merkmalsdimensionen ihrerseits werden an sensorische Eingaben gebunden.

Neben der Verankerung von Symbolen durch sensorisch-motorische Interaktion mit der realen Welt oder durch eine sprachliche Schnittstelle (Vorschläge 2 und 3 von [HARNAD 1990](#)), gibt es auch die Möglichkeit, die Modelle<sup>2</sup> einer Wissensbasis einzuschränken:

„A model theory can determine the actual meaning of the logical symbols of the formal language, but it does not determine the actual meaning of the tokens. The only way to do that is by restricting the set of possible models of the theory, for example by adding axioms. All we can say of a token is that in this model it means this, in that one it means that. There is no single ‚meaning‘ of a formal token (unless there is only a single model): we cannot point to something and say, *that* is the meaning.“ ([HAYES 1985b](#), S. 12f)

Das Zufügen eines neuen Faktes (Axioms) zu einer Wissensbasis führt zu einer Einschränkung der Modelle der Wissensbasis. Dabei ist ein Fakt neu, wenn er nicht aus den

---

<sup>2</sup> „Modell“ im Sinne der Modelltheorie, vgl. [Abschnitt 1.1](#).

bestehenden Fakten abgeleitet werden kann. Je reichhaltiger also die Axiomatisierung der Wissensbasis ist, um so kleiner ist die Anzahl ihrer Modelle. Im Idealfall hat die Wissensbasis nur ein einzelnes Modell und dieses entspricht dem repräsentierten Wissen. Dieser Idealfall ist jedoch nur schwer zu erreichen.

Zusammenfassend lassen sich zwei Aspekte des *Symbol Grounding Problems* bei der Wissensrepräsentation ausmachen: (1) Der Benutzer eines Wissensrepräsentationssystems muss in der Lage sein, die Symbole richtig zu verwenden und zu verstehen. (2) Das Wissensrepräsentationssystem muss in der Lage sein, angemessen zu schlussfolgern. Hinsichtlich beider Aspekte können Ontologien einen Beitrag leisten: Die Angabe natürlichsprachlicher Definitionen ermöglicht dem Benutzer die richtige Symbolverwendung und eine reichhaltige Axiomatisierung beschränkt die möglichen Modelle und unterstützt dadurch die Ableitung adäquater Schlussfolgerungen. Neben dem begrifflichen *Grounding* von Symbolen durch eine Ontologie, können Symbole, die konkrete Objekte repräsentieren, auch durch Verweise (Annotationen) auf entsprechende Datenbankeinträge mit einer Bedeutung versehen werden.

#### 3.2.2. Zum Ontologiebegriff

Bevor informatische Ontologien näher beleuchtet werden, soll zunächst gezeigt werden, wie der originär philosophische Begriff „Ontologie“ in der Informatik Einzug gehalten hat. Dafür wird zunächst kurz die traditionelle philosophische Auffassung von Ontologie dargestellt und anschließend die Übernahme des Begriffs durch die Informatik beschrieben. Neben der Begriffsgenese dient dieser Abschnitt einer Klarstellung des Verhältnisses zwischen philosophischen und informatischen Ontologien.

Zahlreiche Publikationen beschäftigen sich mit der Übernahme des Ontologiebegriffs durch die Informatik (z.B. [SMITH ET AL. 2001](#) und [SMITH 2004](#)), der geschichtlichen Entwicklung innerhalb der Philosophie und der Informatik (z.B. [CORCHO ET AL. 2003](#) und [SOWA 2000a](#)) und dem Verhältnis zwischen philosophischer und informatischer Ontologie (z.B. [GUARINO ET AL. 1995](#) und [GUARINO 1998](#)). Im Folgenden soll auf die wichtigsten Aspekte eingegangen werden.

Die Ontologie als philosophische Disziplin beschäftigt sich mit dem Seienden, d.h. mit den Dingen, die in der Welt existieren. Ein systematischer Zugang zum Seienden wird dabei durch Kategorisierung erreicht, d.h. es werden die grundlegenden Typen von existierenden Dingen untersucht:

„In philosophy 'ontology' (a composition from the Greek words 'onto' (being; that which is) and 'logia' (study, theory)) is the study of existence and reality as such, as well as the categorization of beings and their relations. Practically, it deals with questions like: What entities exist, and how such entities can be classified according to similarities and differences.“ ([HOFMANN 2013](#), S. 60)

Oft werden auch die verschiedenen Kategoriensysteme, die bei diesen Untersuchungen entstehen, als „Ontologien“ bezeichnet (vgl. [GUARINO 1998](#), S. 4). In [SOWA 2000a](#) (Kapitel 2) findet sich eine zusammenfassende Diskussion der wichtigsten Kategorisierungen bedeutender Philosophen, angefangen von Aristoteles über Kant und Hegel bis hin zu Vertretern der Phänomenologie (Husserl und Heidegger) und der klassischen Logik (Peirce und Whitehead). Diesen philosophischen Kategorisierungen werden Kategoriensysteme aus der

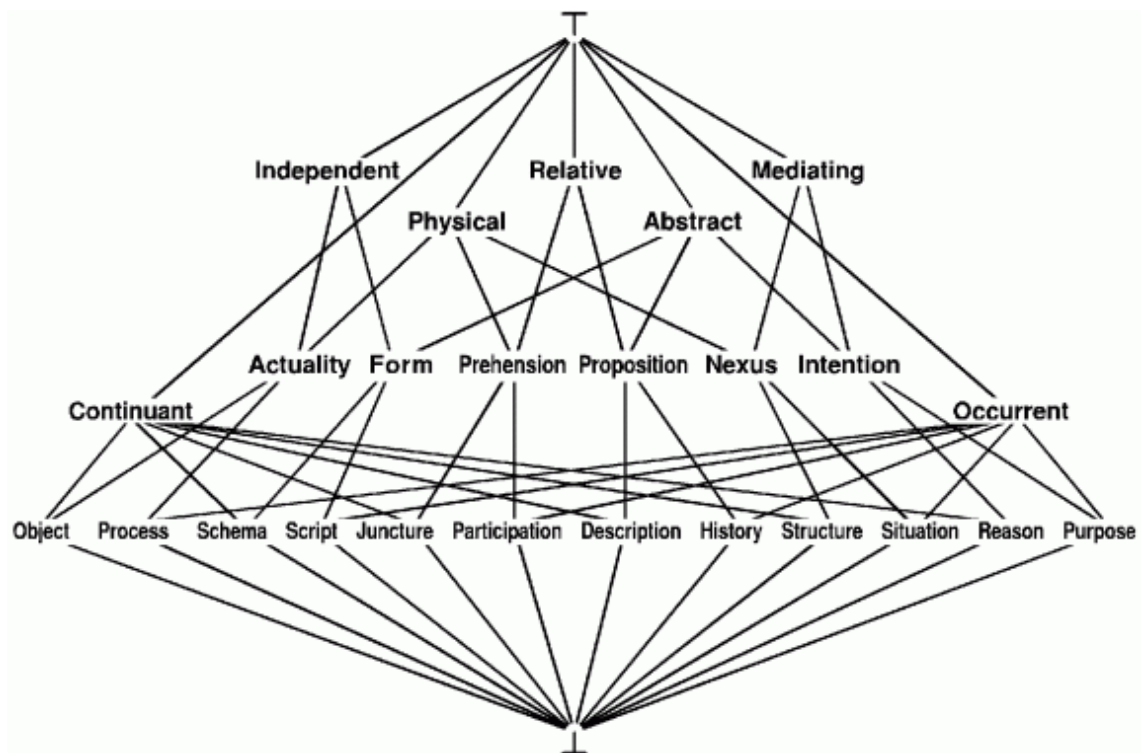


Abbildung 3.2.: Grafische Darstellung des Kategoriensystems von SOWA 2000a als „Sowa-Diamant“ (JANSEN ET AL. 2008, S. 86), Quelle: <http://www.jfsowa.com/figs/bigtop.gif>.

Wissensrepräsentation gegenüber gestellt, etwa diejenige von CYC (LENAT ET AL. 1990). Ausgehend von dieser Diskussion entwickelt SOWA 2000a ein eigenes Kategoriensystem (vgl. auch SOWA 2001), welches als ontologische Grundlage seiner Wissensrepräsentation dient. Die grafische Darstellung dieses Kategoriensystems (Abbildung 3.2) wird auch als „Sowa-Diamant“ bezeichnet (JANSEN ET AL. 2008, S. 86).

Die *formale Ontologie* ist eine philosophische Strömung, die für ontologische Untersuchungen auf formal-logische Mittel zurückgreift:

„Formal ontology is a discipline in which the formal methods of mathematical logic are combined with the intuitive, philosophical analyses and principles of ontology.“ (COCCHIARELLA 2007, S. 3)

Der Begriff der formalen Ontologie geht auf Edmund Husserl zurück (GUARINO ET AL. 1995, S. 26). Hier soll nicht auf die philosophische Kontroverse um den Begriff der formalen Ontologie eingegangen werden (für eine kompakte Darstellung dieser Kontroverse siehe GUARINO ET AL. 1995, S. 26f). Für die folgende Diskussion des Verhältnisses zwischen philosophischer und informatischer Ontologie wird der Begriff der formalen Ontologie im obigen Sinne entsprechend COCCHIARELLA 2007 verwendet, der ihn prägnant auch als „characteristica universalis“ (S. 4) bezeichnet und damit explizit auf Gottfried Wilhelm Leibniz Bezug nimmt.

Anfang der 1990er Jahre taucht der Begriff explizit in der KI auf (z.B. bei [NECHES ET AL. 1991](#) und [GRUBER 1993a](#)). Als Vorläufer dieser Begriffsübernahme in der Wissensrepräsentation kann man [HAYES 1978](#) betrachten, wo der Begriff bereits im Titel („Ontology for Liquids“) auftaucht und damit die (sprachphilosophische) Analyse des Gegenstandsbereichs sowie eine Umsetzung in einen logikbasierten Formalismus verbunden ist. Auch bei [MCCARTHY 1980](#) taucht bereits der Begriff der Ontologie in diesem Sinne auf:

„It was John McCarthy who first recognized the overlap between work done in philosophical ontology and the activity of building the logical theories of AI systems. McCarthy affirmed already in 1980 that builders of logic-based intelligent systems must first ‘list everything that exists, building an ontology of our world’ [23]<sup>3</sup>.“ ([SMITH ET AL. 2001](#), S. v)

Nach [SMITH ET AL. 2001](#) (S. v) hat [HAYES 1985b](#) diese Sichtweise auf die Ontologie von [MCCARTHY 1980](#) übernommen.

Informatische Ontologien werden in [Abschnitt 3.3](#) eingehend diskutiert. Hier reicht zunächst folgende Charakterisierung:

„In computer science (and modeling and simulation) ‘ontologies’ (mind the plural!) are formal specifications of concepts representing entities of a specific knowledge domain and the relationships that can hold between the entities. They intended to be formal descriptions of the so called ‘body of knowledge’ that composes a domain.“ ([HOFMANN 2013](#), S. 60)

An diesem Zitat wird bereits das Verhältnis zwischen philosophischer und informatischer Ontologie deutlich. Insbesondere gibt es eine deutliche Analogie zwischen den formal repräsentierten Kategoriensystemen der formalen Ontologie und den formal spezifizierten Konzepten einer informatischen Ontologie. Ein eklatanter Unterschied besteht jedoch im Grad der Abstraktheit: Während die philosophische Ontologie stets um größte Allgemeingültigkeit bemüht ist, hängen die Konzepte informatischer Ontologien stark vom Wissensbereich und der intendierten Anwendung ab. Dieser Unterschied wird zum Beispiel bei [STOCK 2009](#) (S. 406) deutlich, der Kategorien als die obersten Konzepte (bei [STOCK 2009](#) als „Begriffe“ bezeichnet) in einer „Wissensordnung“ ansieht. Kategorien können hier durchaus im philosophischen Sinne verstanden werden. Neben der Ähnlichkeit in der Form der Resultate gibt es methodische Übereinstimmungen zwischen philosophischer und informatischer Ontologie: Wichtigstes Mittel bei beiden ist die Sprachanalyse.

Die Beziehung zwischen der Aristotelischen Kategorisierung der Welt (vgl. [ARISTOTELES 1998](#)) und den informatischen Ontologien wird ausführlich in [SMITH ET AL. 2008](#) diskutiert. Dabei wird die Vorstellung von Aristoteles verworfen, dass es ein einziges Kategoriensystem gebe. Stattdessen gibt es mehrere Kategoriensysteme, die auf verschiedene Anwendungen zugeschnitten sind. Diese anwendungszentrierte Sicht wird auch als „ontologischer Perspektivismus“ bezeichnet ([SMITH ET AL. 2008](#), S. 21) und folgerichtig von „angewandter Ontologie“ gesprochen. Dieser Perspektivismus liegt auch der in [Abschnitt 1.1.1](#) getroffenen Unterscheidung zwischen Realität und Wirklichkeit zugrunde. [SOWA 2000a](#) bezieht sich bereits mit dem Buchcover auf die Aristotelischen Kategorien: diese werden als „Tree

---

<sup>3</sup>Hierbei handelt es sich um kein wörtliches Zitat aus der Referenz [23] ([MCCARTHY 1980](#)).

of Porphyry“ durch RAMON LULL illustriert. Ganz klar benennt SOWA dann auch die philosophischen Wurzeln seiner Ontologie:

„The KR ontology is a formal ontology based on Whithead’s process philosophy and Peirce’s semiotic that has been designed to serve as a foundation for knowledge representation in databases, knowledge bases, and natural language processing.“ (SOWA 2001)

Der Zusammenhang zwischen formaler Ontologie einerseits und informatischen Ontologien zum *Symbol Grounding* formaler Repräsentationen wird durch das bekannte Zitat von QUINE 1948 (S. 34) auf den Punkt gebracht: „To be is to be the value of a variable“. Danach stellen formale Repräsentationen eine Behauptung darüber auf, was existiert, nämlich die Entitäten, über die die gebundenen Variablen quantifiziert sind (QUINE 1948, S. 34f). Die formalisierte Behandlung dieses „*ontological Commitments*“ formaler Repräsentationen durch CHURCH 1958 kann als Wegbereiter für die informatische Ontologie gesehen werden (vgl. auch GUARINO ET AL. 1994). Ein solches *ontological Commitment* wird prinzipiell von jedem Software- oder Datenbankentwickler mit der Konzeptualisierung getroffen:

„For a database or knowledge base, ontology determines the categories of things that exist or may exist in an application domain. Those categories represent the *ontological commitments* of the designer or knowledge engineer.“ SOWA 2000a (S. 134)

Die Aufgabe informatischer Ontologien ist es, dieses *ontological Commitment* explizit und algorithmisch verfügbar zu machen.

In GUARINO ET AL. 1995 und GUARINO 1998 wird versucht, die Begriffsverwirrung um die „Ontologie“ syntaktisch zu lösen, indem die philosophische Disziplin als „Ontology“ (mit großem 'o', ohne Pluralform) bezeichnet wird und konkrete philosophische Kategoriensysteme und konkrete informatische Artefakte, die Konzepte definieren, als „ontologies“ (kleingeschrieben, sowohl Singular- als auch Pluralform). Im folgenden Text wird auf so eine syntaktische Unterscheidung verzichtet und für die informatische Ontologie meist kurz die Bezeichnung „Ontologie“ verwendet. Im Kontext dieser Arbeit (Wissensrepräsentation) sollte dies außerhalb des vorliegenden Abschnitts nicht zu einer Begriffsverwirrung führen.

Für eine Vertiefung des Zusammenhangs zwischen philosophischer und informatischer Ontologie sei nochmals auf SMITH 2004 verwiesen. Ausgehend von der *beschreibenden* Ontologie der Philosophie (S. 155f), über die formale Beschreibung des „*ontological commitments*“ (S. 156f) wissenschaftlicher Theorien gelangt SMITH 2004 schließlich zu den „*conceptualizations*“ (S. 161f) in der Informatik und beschreitet dabei einen ähnlichen Weg, wie die Darstellung in diesem Abschnitt.

#### 3.2.3. Logische Präzision und ästhetische Prägnanz von Ontologien

Es gibt zwei extreme Positionen, mit Hilfe von Ontologien das *Symbol Grounding Problem* anzugehen: Zum einen können durch logische Axiomatisierung die möglichen (intrinsischen) Interpretationen von Ausdrücken eingeschränkt werden. Dadurch wird automatisches Schlussfolgern auf dem repräsentierten Wissen möglich. Allerdings ist eine vollständige logisch-axiomatische Festlegung der Symbolbedeutung nicht möglich. Dem entgegengesetzt erhalten Ausdrücke durch Einbeziehung menschlicher Akteure eine (extrinsische)



Interpretation. Dabei wird Symbolen durch außerlogische Referenzen (natürlichsprachliche Beschreibungen, Bilder, Verweise auf konkrete Dinge in der Welt) eine Bedeutung zugewiesen, die innerhalb einer Gemeinschaft akzeptiert ist, d.h. die Beschreibungen erlauben ein „shared understanding“ (USCHOLD ET AL. 1996, S. 1) der Symbole. Damit geht eine Standardisierung der Symbolverwendung einher. Die Bedeutung rein außerlogisch definierter Symbole ist einer Computerverarbeitung jedoch nicht zugänglich.

Hier wird die Auffassung vertreten, dass das *Symbol Grounding Problem* nur durch eine Integration beider Ansätze zufriedenstellend gelöst werden kann. Ontologien in einem solchen Sinne gestatten aufgrund der Dualität ihrer Interpretation eine Verbindung von formalsprachlichen Symbolen und realweltlichen Bedeutungen. Die Definition von Ontologie, wie sie von GRUBER 1993a gegeben wird (siehe Abschnitt 3.3.1), wird somit um die explizite Forderung nach natürlichsprachlichen Beschreibungen erweitert. Ontologien erscheinen damit als Schnittstellen zwischen natürlichsprachlichen Ausdrücken und formalsprachlichen Entitäten. Die formalsprachlichen Entitäten codieren somit terminologisches Wissen. Unter den Prinzipien für die Wissensrepräsentation in SOWA 2000a (S. 134ff, zurückgehend auf DAVIS ET AL. 1993) findet sich eine entsprechende Forderung: Ein Wissensrepräsentationssystem soll ein Medium für effiziente Berechnungen sein (Prinzip 4) und ein Mittel menschlicher Kommunikation darstellen (Prinzip 5).

Inspiziert von einer Diskussion über Definitionen in GABRIEL 2010 lässt sich die formale Festlegung der Symbolbedeutung in Ontologien als *logische Präzision* bezeichnen. Die natürlichsprachlichen Festlegungen der Symbolbedeutung betreffen entsprechend die *ästhetische Prägnanz* einer Ontologie. Eine Ontologie ist somit eine Sammlung von logisch präzisen und ästhetisch prägnanten Konzeptdefinitionen, die die Bedeutung von Symbolen festlegen.

Nach GABRIEL geht es darum, Begriffe (Konzepte) „klar und deutlich“ zu definieren, wobei diese Unterscheidung durch Leibniz getroffen wurde:

„Klar sind danach solche Begriffe, die es erlauben, die unter sie fallenden Gegenstände wiederzuerkennen.[...] [D]eutlich sind solche klaren Begriffe, für die eine Definition (im Sinne der Zerlegung des Begriffs in seine Merkmale) vorliegt.“ (GABRIEL 2010, S. 378)

Ästhetische Prägnanz führt demnach zu klaren Begriffen, logische Präzision zu deutlichen Begriffen. Dies ist genau die Forderung an Ontologien: Der menschliche Benutzer soll sich darüber im *Klaren* sein, was mit einem Symbol gemeint ist und das Wissensrepräsentationssystem soll mit Hilfe der entsprechenden Begriffsmerkmale dazu passende Schlussfolgerungen ziehen können.

### 3.3. Informatische Ontologien

Nachdem im letzten Abschnitt gezeigt wurde, wie informatische Ontologien zum *Grounding* von Symbolen benutzt werden können und welche Eigenschaften eine Ontologie haben muss, um dem gerecht zu werden, soll im Folgenden konkreter auf informatische Ontologien und deren Realisierung als informatische Artefakte eingegangen werden. Dafür wird zunächst die Bandbreite informatischer und informationstechnischer Artefakte beleuchtet, die unter dem Begriff „Ontologie“ eingeordnet werden und eine Begriffsbestimmung vorgenommen



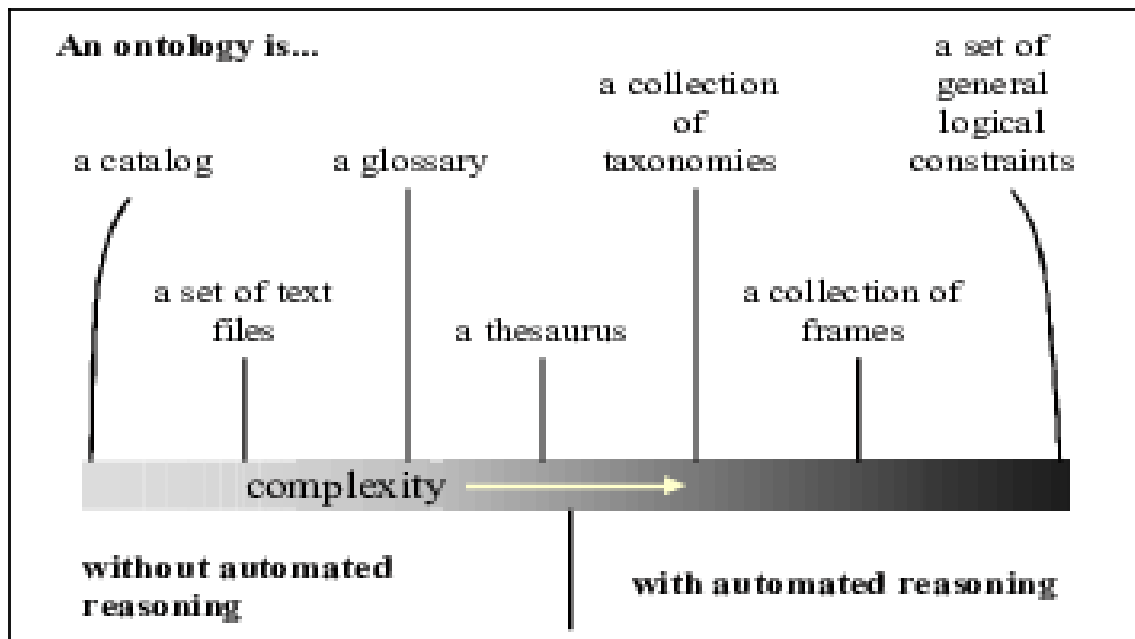


Abbildung 3.3.: Spektrum informatischer Ontologien. Aus [SMITH ET AL. 2001](#) (S. v).

([Abschnitt 3.3.1](#)). Anschließend wird dargestellt, wie informatische Ontologien beschaffen sind und wie sie technisch realisiert werden ([Abschnitt 3.3.2](#)). Für die Entwicklung einer Ontologie ist ein ingenieurwissenschaftliches Vorgehen notwendig, das zudem in einer kollaborativen Weise von mehreren Entwicklern und Anwendern gemeinsam gestaltet wird. Damit beschäftigt sich abschließend [Abschnitt 3.3.3](#).

### 3.3.1. Begriffsbestimmung informatischer Ontologie

Eine Vielfalt von informatischen Artefakten wird als „Ontologie“ bezeichnet. Zunächst werden in diesem Abschnitt kurz die unterschiedlichen Formen dieser Artefakte und die Rolle, die sie in Informationssystemen spielen, dargestellt. Dabei zeigt sich, dass der Begriff „Ontologie“ in der Informatik sehr unterschiedlich verwendet wird. Ausgehend von diesen unterschiedlichen Auffassungen werden Versuche einer Begriffsbestimmung für die informatische Ontologie diskutiert.

Das breite Spektrum an Artefakten, die als informatische Ontologien angesehen werden, wird in [SMITH ET AL. 2001](#) in einer Übersicht dargestellt (vgl. [Abbildung 3.3](#)) und die Bandbreite der Begriffsverwendung von „Ontologie“ hinterfragt. Die beiden Seiten in [Abbildung 3.3](#) werden auch häufig mit den Begriffen „lightweight“ (links) und „heavyweight“ (rechts) Ontologie bezeichnet ([FERNÁNDEZ-LÓPEZ ET AL. 2004](#), S. 334), um den Grad der Formalisierung auszudrücken. Dieselbe Unterscheidung wird häufig auch als „informale“ vs. „formale“ Ontologien bezeichnet (etwa [SOWA 2000a](#), S. 492). In diesem Sinne sind logisch präzise Ontologien „heavyweight“.

Im Folgenden werden die wichtigsten Artefakte beschrieben, die als Ontologien bezeichnet werden. Die Artefakte sind dabei nach steigendem Grad der Formalisierung angeführt:

**Kontrollierte Vokabulare (Thesauren)** Eine stark verbreitete Auffassung, insbesondere in der Bioinformatik, sieht Ontologien als (standardisierte) Listen von Wörtern, auf die man sich für die Beschreibung von Dingen eines Wissensgebiets einigt.

**Taxonomien** In einer Taxonomie werden Konzepte entlang einer Hierarchie von Arten und Unterarten angeordnet. Diese über eine pure Auflistung von Wörtern hinausgehende Struktur erlaubt bestimmte Schlussweisen.

**Terminologien** Festlegung der Wörter (Termini oder englisch „terms“) eines Fachgebietes und deren Bedeutung. Meist wird die Bedeutung der Wörter durch natürlichsprachliche Definitionen gegeben, die einer computergestützten Verarbeitung nur begrenzt zugänglich sind. Erfolgt ausschließlich eine natürlichsprachliche Beschreibung der Wörter, liegt der Spezialfall eines kontrollierten Vokabulars vor. Sind die Definitionen im Sinne Aristoteles (Oberkonzept und Unterscheidungsmerkmale) angelegt, handelt es sich um eine Taxonomie.

**Konzeptualisierungen** Im Gegensatz zu den vorherigen Ansätzen, bei denen Wörter im Mittelpunkt stehen, sind hier Konzepte und Beziehungen zwischen diesen zentral. Dies hat entscheidende Vorteile, insbesondere bezüglich möglicher Schlussfolgerungen, der Semantik zusammengesetzter Konzepte und der kontextabhängigen Semantik von Wörtern.

In [USCHOLD ET AL. 1999](#) findet sich eine Klassifikation von Verwendungszwecken von Ontologien und eine Darstellung zugehöriger Anwendungsszenarien. Dabei wird eine sehr breite Auffassung von Ontologie zugrunde gelegt:

„An ONTOLOGY may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the DOMAIN and constrain the possible interpretations of terms.“ ([USCHOLD 1998](#), S. 12, Hervorhebung wie im Original)

Unterschiedliche Auffassungen des Begriffs „Ontologie“ werden in [GUARINO ET AL. 1995](#) betrachtet. Die aktuelle Diskussion der Begriffsverwendung in [CORCHO ET AL. 2003](#) (S. 42ff) folgt dieser Betrachtung. Die wohl prominenteste Definition informatischer Ontologien durch GRUBER führt diese auf Konzeptualisierungen zurück:

„An ontology is an explicit specification of a conceptualization.“ ([GRUBER 1993a](#), S. 1)

Diese Definition wird auch in den meisten aktuellen Publikationen zum Thema zitiert und hat sich als Quasi-Standard etabliert. Und dies, obwohl zahlreiche Probleme an GRUBERS Definition erkannt und mehrfach Vorschläge zu einer besseren Definition gemacht wurden (vgl. z.B. [GUARINO ET AL. 1995](#) und [SMITH 2008](#)). Nach [SMITH ET AL. 2001](#) (S. v) lässt GRUBERS Definition zu viel Raum für Interpretationen. Eine solch breite Definition ist jedoch notwendig, um das weite Spektrum von informatischen Artefakten, die als „Ontologien“ klassifiziert werden, zu umfassen.

Die Definition von [GRUBER 1993a](#) fasst eine Ontologie als ein informatisches Artefakt („explicit specification“) auf, welche in einer geeigneten formalen Sprache codiert

ist. Demnach müssten zwei verschiedene Codierungen derselben Konzeptualisierung als zwei unterschiedliche Ontologien angesehen werden. Um diese zu vermeiden, sollte die Konzeptualisierung selbst als Ontologie angesehen werden, wie etwa in folgendem Zitat:

„Ontologies are content theories about the sorts of objects, properties of objects, and relations between objects that are possible in a specified domain of knowledge. They provide potential terms for describing our knowledge about the domain.“ (CHANDRASEKARAN ET AL. 1999, S. 20)

Natürlich muss eine solche Konzeptualisierung explizit spezifiziert und in einer entsprechenden formalen Sprache codiert werden, um in Wissensrepräsentationssystemen eingesetzt werden zu können. GRUBERS Definition müsste also entsprechend modifiziert werden zu:

Eine Ontologie ist eine Konzeptualisierung eines Wirklichkeitsausschnitts, die in einer formalen Sprache explizit spezifiziert wird.

Ontologien sind also in dieser Sichtweise Konzeptualisierungen, an die jedoch der Anspruch gestellt wird, dass sie explizit formuliert sind.

Was ist nun aber eine Konzeptualisierung eines Wirklichkeitsausschnitts? Der letzte Satz des obigen Zitats von CHANDRASEKARAN ET AL. 1999 benennt den Zweck einer Konzeptualisierung: Sie stellt die Grundelemente (Terme) für Aussagen über den Wirklichkeitsausschnitt zur Verfügung. Wie bereits aus dem Wortstamm deutlich wird, stellt eine Konzeptualisierung also die relevanten Konzepte bereit, um über einen Wirklichkeitsausschnitt zu reden. Für die Newtonsche Mechanik wären das z.B. Konzepte wie Ort, Masse, Kraft und Impuls. Damit ist eine Konzeptualisierung etwas, auf das zum *Symbol Grounding* referenziert werden kann.

GRUBER fasst eine Konzeptualisierung extensional auf, d.h. durch Angabe der Extensionen der verwendeten Relationen. Während bei einer extensionalen Konzeptualisierung die Objektinstanzen aufgezählt werden, die in einer bestimmten Relation stehen, wird bei einer intensionalen Konzeptualisierung versucht, das Wesen eines Konzeptes zu erfassen, unabhängig von der speziellen Realisierung dieses Wesens in konkreten Instanzen. Tatsächlich ist es bei der Formulierung einer intensionalen Konzeptualisierung nicht nötig, überhaupt Instanzen zu erwähnen. Es werden nur die Extensionen von Konzepten mit Hilfe von Axiomen partiell definiert (eingeschränkt).

Eine extensional festgelegte Konzeptualisierung ist auf einen bestimmten Zustand der Wirklichkeit (mögliche Welt) beschränkt. Diese Auffassung wird von GUARINO ET AL. 1995 kritisiert. Stattdessen soll eine Konzeptualisierung ja gerade von den konkreten Zuständen der Wirklichkeit abstrahieren. Dies gelingt intensional durch formale *Constraints* („Axiome“ in der Sprache der Logik, vgl. Abschnitt 3.1.1), welche die möglichen Extensionen beschränken (GUARINO ET AL. 1995, S. 29). Eine solche intensionale Spezifikation legt im Allgemeinen nicht eindeutig eine Konzeptualisierung fest. Stattdessen werden mögliche Konzeptualisierungen charakterisiert, die der Spezifikation genügen, es wird also ein *ontological Commitment* spezifiziert (GUARINO ET AL. 1995).

Die Versuche einer Begriffsbestimmung für informatische Ontologien bewegen sich zwischen zwei extremen Standpunkten, für die stellvertretend die beiden folgenden Zitate dienen sollen. Zum einen wird die Ontologie als Sammlung von Konzeptbezeichnern gesehen, zwischen welchen formale Beziehungen spezifiziert werden: „a single consistent, stable, and

highly expressive set of category labels“ (SMITH 2004, S. 159). Die Konzeptbezeichner („category labels“) stehen dabei für die repräsentierten Konzepte. Eine nähere Bestimmung dieser Konzepte, etwa in Form von Definitionen, erfolgt dabei nicht. Diese Auffassung verlangt von einer Ontologie logische Präzision, verzichtet aber auch ästhetische Prägnanz (vgl. Abschnitt 3.2.3). Zum anderen wird nur auf die extrinsische Bedeutung der Konzepte Bezug genommen und eine Ontologie schwammig als „explicit account of a shared understanding“ (USCHOLD ET AL. 1996, S. 1) bezeichnet. Damit wird die ästhetische Prägnanz als das entscheidende Merkmal von Ontologien angesehen. In Abschnitt 3.2.3 wird argumentiert, dass eine Ontologie beide Sichtweisen vereinen muss, um als *Symbol Grounding* für Wissensrepräsentationssysteme dienen zu können und damit zum „Herz“ eines solchen Systems zu werden:

„Given a domain, its ontology forms the heart of any system of knowledge representation for that domain.“ (CHANDRASEKARAN ET AL. 1999, S. 21)

Die Beziehungen zwischen den Termen in einer Ontologie und ihren extrinsischen Bedeutungsinhalten bringt folgendes Zitat noch einmal auf den Punkt:

„First of all, ontology is a representation vocabulary, often specialized to some domain or subject matter. More precisely, it is not the vocabulary as such that qualifies as an ontology, but the conceptualizations that the terms in the vocabulary are intended to capture.“ (CHANDRASEKARAN ET AL. 1999, S. 20)

Ein anhaltender Streit betrifft die Frage, ob eine informatische Ontologie die reale Welt repräsentieren muss oder „nur“ eine spezifische Auffassung derselben, also eine Wirklichkeit (für die Unterscheidung zwischen Realität und Wirklichkeit siehe Abschnitt 1.1.1). Die Definition von GRUBER enthält keinen direkten Verweis auf die reale Welt und ist somit ein Beispiel für letztere Auffassung. Diese Auffassung wird teilweise heftig kritisiert, z.B. durch SMITH 2004 (S. 160ff). Hier soll nicht weiter auf diesen Realismusstreit eingegangen und stattdessen auf die Literatur verwiesen werden. Aktuelle Diskussionen finden sich z.B. in SMITH 2008, DUMONTIER ET AL. 2010 und SCHÄUFLER 2012. In dieser Arbeit wird ein pragmatischer nicht-realistischer Standpunkt eingenommen, wie er sich auch in folgendem Zitat zeigt:

„But I now believe that such naive ontological realism is unfounded. In the actual world, there are no a priori individuals at all. [...] [D]ivisions of it [the universe – Anmerkung des Autors] into conceptual individuals are made by us, by our language, not by nature.“ (HAYES 1985a, S. 106)

Letztendlich muss sich eine informatische Ontologie daran orientieren, welche Konzepte in dem zu repräsentierenden Wissen vorkommen, unabhängig davon, ob diese Konzepte Klassen real existierender Gegenstände bezeichnen oder nicht. Der Bezug auf die beschriebene Wirklichkeit kann dabei hilfreich sein, muss aber nicht.

Oft werden Ontologien nach ihrem Abstraktionsgrad unterschieden: *Top-Level Ontologien* (auch „upper-level ontologies“) beschreiben generelle Konzepte, wie Raum und Zeit, Objekt und Prozess, während *Domainontologien* ein spezielles Wissensgebiet begrifflich erfassen (siehe etwa GUARINO 1998, S. 9f).

Die hier vertretene Auffassung informatischer Ontologie lässt sich in folgender Definition zusammenfassen:

Eine *informatische Ontologie* ist eine Konzeptualisierung eines Wirklichkeitsausschnitts, die in einer formalen Sprache explizit spezifiziert ist. Dabei sind die Konzepte logisch-präzise definiert und mit ästhetisch-prägnanten Beschreibungen versehen.

Für die Verwendung in Computerprogrammen muss eine solche informatische Ontologie in einer geeigneten formalen Sprache codiert werden. Damit beschäftigt sich der folgende Abschnitt.

### 3.3.2. Spezifikation und Codierung informatischer Ontologien

Die bisherige Darstellung informatischer Ontologien ist noch ziemlich abstrakt. In diesem Abschnitt soll nun beschrieben werden, wie informatische Ontologien konkret aufgebaut sind und wie sie spezifiziert und in entsprechenden formalen Sprachen codiert werden.

Informatische Ontologien beschreiben logisch präzise und ästhetisch prägnant *Konzepte* einer Wissensdomäne (vgl. [Abschnitt 3.2.3](#)). Konzepte werden dabei als Klassen von Objekten mit gemeinsamen Eigenschaften aufgefasst. Die Objekte, die unter ein Konzept fallen, heißen auch seine *Instanzen*. Die Menge aller Instanzen eines Konzeptes wird seine *Extension* (auch „Begriffsumfang“) genannt. Die *Intention* eines Konzeptes, sein Begriffsinhalt, ist die Menge aller charakteristischen Merkmale dieses Konzeptes.

Die Merkmale eines Konzeptes können als Relationen zu anderen Konzepten oder als Beschränkungen von Eigenschaften auftreten. Dabei kommen Eigenschaften, wie etwa „Gewicht“ den Instanzen zu und können qualitativer (z.B. „schwer“) oder quantitativer (z.B. 100 kg) Ausprägung sein. Die wichtigste Relation zwischen Konzepten ist die *Subsumtion*. Ein Konzept A wird unter ein Konzept B subsumiert, wenn jede Instanz von A auch eine Instanz von B ist. Andere Sprechweisen sind „B subsumiert A“ oder „B ist Oberkonzept von A“ bzw. „A ist Unterkonzept von B“ oder kurz „A ist ein B“. Letztere Formulierung ist auch die Herkunft der Relation *is-a*, die häufig in Ontologien Verwendung findet. Ebenso häufig ist die Schreibweise *subclass-of*. Durch Subsumtionsbeziehungen wird eine Konzepthierarchie aufgebaut. Die Subsumtionshierarchie bildet das „Rückgrat“ jeder Ontologie.

Eine Subsumtionsbeziehung zwischen zwei Konzepten kann entweder explizit vorgegeben werden oder aus anderen Merkmalen durch eine Inferenzmaschine geschlussfolgert werden. Explizit vorgegebene Merkmale werden auch als „asserted“ bezeichnet, geschlussfolgerte entsprechend als „inferred“.

Neben der Subsumtion kann es noch weitere Relationen zwischen Konzepten geben. Häufig anzutreffen sind etwa partonomische Relationen (z.B. *part-of*). Meistens sind die Relationen auf der Konzeptebene durch Relationen auf der Instanzebene definiert. So kann z.B. „Finger *part-of* Hand“ interpretiert werden als: „Für jede Instanz eines Fingers gibt es eine Instanz einer Hand, deren Teil der Finger ist.“ Dabei ist „Teil-sein“ die entsprechende Relation auf Instanzebene. Für eine ausführliche Darstellung der Rückführung gebräuchlicher Konzeptrelationen auf Instanzrelationen siehe [SMITH ET AL. 2005](#).

Die Spezifikation einer Ontologie muss folgende Anforderungen erfüllen:

**Identifikation eines Konzeptes** Jedes Konzept ist durch eine Benennung (Identifikator, kurz ID) eindeutig referenzierbar. Die sogenannte *Unique Name Assumption* muss

dabei übrigens nicht gelten, d.h. es kann durchaus unterschiedliche IDs für ein Konzept geben.

**Bezeichnung eines Konzeptes** Jedes Konzept besitzt mindestens eine natürlichsprachliche Bezeichnung. Bei mehrsprachigen Ontologien kann es unterschiedliche Bezeichnungen in verschiedenen Sprachen geben. Jeder Bezeichnung ist dann eine Sprache zuzuordnen.

**Subsumtion** Für ein Konzept werden die Oberkonzepte und Unterkonzepte explizit oder implizit angegeben.

**Relationen** Für ein Konzept sind unterschiedliche Relationen zu anderen Konzepten festzulegen.

**Eigenschaften** Einem Konzept werden geeignete Eigenschaften zugeordnet.

**Definitionen** Ein Konzept ist möglichst vollständig durch eine natürlichsprachliche Definition zu beschreiben. Diese Definition sollte verträglich mit den Subsumtionsbeziehungen sowie den Relationen und Eigenschaften sein. Idealerweise werden Definitionen im Sinne von Aristoteles mit Hilfe des nächstliegenden Oberbegriffs („genus proximum“) und charakteristischer Unterschiede („differentia specifica“) angegeben (vgl. STOCK 2009, S. 407). Das entsprechende Oberkonzept sollte sich in einer Subsumtionsbeziehung widerspiegeln und die Unterscheidungsmerkmale durch Relationen und Eigenschaften repräsentiert werden. Ein solches Vorgehen wird z.B. in MICHAEL ET AL. 2001 für eine anatomische Ontologie beschritten: Die aristotelischen Definitionen spiegeln sich in der formalen Spezifikation der Ontologie wider.

Wie oben dargelegt, beschreiben informatische Ontologien Konzepte und Beziehungen zwischen Konzepten. Im Folgenden wird gezeigt, wie informatische Ontologien spezifiziert und computerverarbeitbar codiert werden. Für die Spezifikation, Codierung und Nutzung von Ontologien ist folgende informatische Infrastruktur notwendig, auf die nachfolgend im Einzelnen eingegangen wird:

- formale Sprachen zur Codierung der Ontologien,
- Inferenzmaschinen für die jeweiligen formalen Sprachen,
- Entwicklungswerkzeuge für Ontologien und
- Zugriffsmechanismen für die spezifizierten Ontologien.

#### Formale Sprachen

Es gibt zahlreiche logikbasierte formale Sprachen für die Codierung informatischer Ontologien, wie etwa KIF (*Knowledge Interchange Format*, GENESERITH ET AL. 1992), OBO Format ([http://www.geneontology.org/G0.format.obo-1\\_2.shtml](http://www.geneontology.org/G0.format.obo-1_2.shtml)) und OWL. Klassische Ansätze wie Semantische Netze und Frames wurden aufgrund bestehender Defizite zu sogenannten Beschreibungslogiken weiterentwickelt (vgl. NARDI ET AL. 2003, dort findet sich auch eine gute Einführung in Beschreibungslogiken).



Beschreibungslogiken besitzen eine wohldefinierte Semantik und verfügen über eine ausgebauten Theorie entsprechender Schlussfolgerungsalgorithmen und deren Berechnungskomplexität. Beschreibungslogiken stellen den State of the Art logikbasierter Sprachen für die Wissensrepräsentation dar.

In Beschreibungslogiken wird zwischen der Repräsentation von terminologischem Wissen (in der sogenannten **TBox**, *Terminological Box*) und von Faktenwissen (in der sogenannten **ABox**, *Assertional Box*) unterschieden. Die **TBox** kann für die Spezifikation von Ontologien benutzt werden, in der **ABox** können Fakten mit Hilfe der Ontologie formuliert werden.

Für die Formulierung biologischer Ontologien wird häufig das **OBO Format** verwendet. Mit dem **OBO Format** können jedoch nur sehr einfache Zusammenhänge ausgedrückt werden. Außerdem existieren keine Inferenzmaschinen für das **OBO Format**. Aus diesen Gründen setzt sich mehr und mehr die Sprache **OWL** durch, welche eine verbreitete Realisierung einer Beschreibungslogik darstellt. Im Folgenden wird näher auf **OWL**<sup>4</sup> eingegangen, weil diese Sprache zur Spezifikation der Ontologie **TEDDY** verwendet wird, welche in **Kapitel 6** vorgestellt wird. Dabei sollen nur die wichtigsten Prinzipien und Konstrukte von **OWL** dargestellt und an Beispielen illustriert werden. Für Details siehe die entsprechende Dokumentation, einen guten Einstieg bietet etwa **HITZLER ET AL. 2012**.

**OWL** ist eine vom **W3C** (*World Wide Web Consortium*, <http://www.w3.org/>) entwickelte formale Sprache für die Spezifikation von Ontologien, deren Ursprünge auf die Idee des *Semantic Web* zurückgehen. Als für das Internet konzipierte Sprache wird jede in **OWL** beschriebene *Entität* (Klasse, Eigenschaft und Individuum) mittels einer **IRI** (*Internationalized Resource Identifier*) identifiziert. Die **IRI** stellt also die ID der entsprechenden Entität dar. Die **IRIs** einer Ontologie werden in der Regel in einem Namensraum zusammengefasst, welcher als *Ontology IRI* bezeichnet wird (**MOTIK ET AL. 2012**, Abschnitt 3.1). Verschiedene Ontologien sollten unterschiedliche Namensräume besitzen. Der Namensraum von **TEDDY** ist z.B. <http://identifiers.org/teddy/>, eine typische **TEDDY**-ID entsprechend: [http://identifiers.org/teddy/TEDDY\\_0000000](http://identifiers.org/teddy/TEDDY_0000000).

Es existieren zahlreiche Dateiformate für die Serialisierung von **OWL**, also konkrete Darstellungsformen für die abstrakte **OWL**-Syntax. Der Quasistandard für den Austausch von Ontologien im Internet und zwischen Werkzeugen ist das **XML**-basierte Dateiformat **RDF/XML** (**BECKETT 2004**), in welchem auch **TEDDY** serialisiert wird. Ein Dateiformat, welches für Menschen besser lesbar ist, ist die **Manchester Syntax** (**HORRIDGE ET AL. 2012**). Diese wird häufig in Tutorien über **OWL** und Diskussionen über konkrete Ontologien eingesetzt und auch in dieser Arbeit verwendet.

Die Entitäten, d.h. Individuen (Objekte), Konzepte (Klassen) und Relationen (Eigenschaften), stellen die Grundbausteine eines **OWL**-Dokumentes dar. Für die Spezifikation von Ontologien sind insbesondere die Klassen (**Class**) und die Eigenschaften (**Property**) von Bedeutung, mit denen die Konzepte und deren Relationen untereinander spezifiziert werden können. Für die Subsumtionsbeziehung zwischen Konzepten steht die entsprechende Relation **SubClassOf** zur Verfügung. Die Oberklasse aller Klassen einer Ontologie ist **Thing**.

Eine Beziehung zwischen zwei Individuen kann mit einer geeigneten Objekteigenschaft

---

<sup>4</sup>Eigentlich handelt es sich bei **OWL** um eine Familie formaler Sprachen. Die aktuelle Variante ist **OWL2**, welche auch im Folgenden beschrieben wird. Die Unterschiede sind im Kontext dieser Arbeit jedoch irrelevant. Aus diesem Grund wird hier kurz von „**OWL**“ gesprochen.



(`ObjectProperty`) ausgedrückt werden. Eine Eigenschaft eines Individuums kann durch eine entsprechende Dateneigenschaft (`DataProperty`) formalisiert werden, deren Werte Literale eines bestimmten Datentyps (z.B. Ganzzahl, Gleitkommazahl, [IRI](#), String) darstellen. Für erlaubte Datentypen siehe [MOTIK ET AL. 2012](#) (Abschnitt 4).

Mit diesen Eigenschaften für Individuen können durch Axiome Relationen zwischen Konzepten definiert werden. Axiome können entweder Äquivalenz (`EquivalentClasses`) oder Subsumtion (`SubClassOf`) zwischen Klassen ausdrücken. Für die Spezifikation von Relationen zwischen Konzepten werden in solchen Axiomen sogenannte Klassenausdrücke verwendet. Klassenausdrücke sind induktiv definiert (für Details siehe [HITZLER ET AL. 2012](#)). Dies soll an einigen Beispielen aus [HITZLER ET AL. 2012](#) verdeutlicht werden. Auf eine streng formale Darstellung wird zugunsten des Verständnisses der Grundideen verzichtet.

Das folgende Axiom definiert die Klasse `Mother` als Menge derjenigen Individuen, die sowohl Instanz der Klasse `Woman` als auch der Klasse `Parent` sind:

```
Class: Mother
  EquivalentTo: Woman and Parent
```

Eine Mutter ist also eine Frau und ein Elternteil. Semantisch gesehen handelt es sich bei der Klasse `Mother` also um den Durchschnitt der Klassen `Woman` und `Parent`. Dabei werden [OWL](#)-Klassen als Mengen von Individuen interpretiert, `and` entsprechend als Mengendurchschnitt. Analog gibt es die Vereinigung, z.B. drückt

```
Class: Parent
  EquivalentTo: Mother or Father
```

aus, dass ein Elternteil Mutter *oder* Vater ist.

Die oben beschriebenen Eigenschaften für Individuen können auf verschiedene Weise in Klassenausdrücken verwendet werden. Ein Elternteil kann etwa als Person definiert werden, der mindestens ein Kind besitzt:

```
Class: Parent
  EquivalentTo: hasChild some Person
```

Neben der existentiellen (`some`) gibt es auch die universelle Quantifizierung über Eigenschaften (`only`) sowie die Kardinalitätsbeschränkung (`min`, `max`, `exactly`), mit welcher die Anzahl an Füllern für die entsprechende Eigenschaft beschränkt werden kann.

Ähnliche Axiome können mit Dateneigenschaften formuliert werden. Ein Teenager etwa sollte ein entsprechendes Alter haben (Vergleichszeichen korrigiert gegenüber [HITZLER ET AL. 2012](#)):

```
Class: Teenager
  SubClassOf: hasAge some integer[> 12 , <= 19]
```

Bei diesem Axiom handelt es sich im Gegensatz zu den vorhergehenden um eine partielle Definition (`SubClassOf`), d.h. nicht jedes Individuum mit einem Alter zwischen 13 und 19 Jahren muss ein Teenager sein, wie etwa ein 15-jähriges Auto.

Für die Spezifikation ästhetisch prägnanter Ontologien erlaubt [OWL](#), Entitäten mit Annotationen (`AnnotationProperty`) zu versehen. Die folgenden typischen Annotationseigenschaften in [OWL](#)-Ontologien greifen auf die Vokabulare [RDFS](#) (*Resource Description*

*Framework Schema*, GUHA ET AL. 2004) und SKOS (*Simple Knowledge Organization System*, MILES ET AL. 2009) zurück (gekennzeichnet durch die entsprechenden Namensraum-Präfixe `rdfs` und `skos`):

**Bezeichner** Ein natürlichsprachlicher Bezeichner<sup>5</sup> für ein Konzept kann mit einer geeigneten Annotation vergeben werden. Meist wird hierfür die Eigenschaft `rdfs:label` verwendet.

**Synonym** Alternative Bezeichner werden häufig mit `skos:altLabel` angegeben.

**Referenz** Externe Referenzen für ein Konzept, wie z.B. Quellenangaben, können durch `rdfs:seeAlso` gegeben werden.

**Kommentar** Zusätzliche Kommentare können als `rdfs:comment` gegeben werden.

**Definition** Eine natürlichsprachliche Definition kann durch `skos:definition` angegeben werden.

Auch in TEDDY (siehe Kapitel 6) werden diese Annotationseigenschaften verwendet.

Ein wichtiger Mechanismus in OWL ist der Import. Dabei können Konzepte bestehender Ontologien für die Spezifikation neuer Konzepte verwendet werden. Da jede Ontologie einen eigenen Namensraum besitzt und die Konzepte über IRIs der entsprechenden Namensräume referenziert werden, kann es zu keinen Namenskonflikten bei Ontologie-Import kommen. Der Import verwendet zum einen bestehende Ontologien wieder und ist zum anderen wichtige Voraussetzung für die Interoperabilität von ontologiebasierten Systemen.

### Inferenzmaschinen

Der Aufwand, eine Ontologie logisch präzise zu spezifizieren, zahlt sich aus, wenn die Ontologie für automatische Schlussfolgerungen verwendet wird. Solche Schlussfolgerungen werden durch Inferenzmaschinen (englisch *Reasoner*) realisiert. Die zentralen Schlussfolgerungen für Beschreibungslogiken sind der Subsumtionstest, welcher bestimmt, ob ein Konzept Oberkonzept eines anderen Konzeptes ist, und der Erfüllbarkeitstest, welcher überprüft, ob ein gegebenes Konzept Instanzen haben kann (vgl. NARDI ET AL. 2003, S. 13). Der Erfüllbarkeitstest ist ein spezieller Subsumtionstest danach, ob das leere Konzept Oberkonzept für das gegebene Konzept ist. Ein weiterer Test bezüglich der ABox ist der Instanztest, welcher verifiziert, ob eine gegebene Instanz zu einem gegebenen Konzept gehört (NARDI ET AL. 2003, S. 19). Für OWL existieren ausgereifte Inferenzmaschinen, wie z.B. HermiT (MOTIK ET AL. 2009), Pellet (<http://clarkparsia.com/pellet>) und FaCT++ (<http://owl.man.ac.uk/factplusplus/>).

Zwei Aspekte sind bei Schlussfolgerungen über Ontologien zu beachten: Zum einen stellt die Subsumtionshierarchie eine echte Vererbungshierarchie dar: Axiome eines Konzepts gelten auch in allen Unterkonzepten (MICHAEL ET AL. 2001, S. 1). Zum anderen gilt – zumindest in OWL – die sogenannte „*Open World Assumption*“: Aus der Tatsache, dass etwas *nicht* in der Ontologie steht, folgt *nicht*, dass es nicht zutrifft. Die *Open World*

---

<sup>5</sup>Obwohl die IDs in den obigen Beispielen aus Gründen der Lesbarkeit natürlichsprachlich gewählt wurden (etwa *Mother*), dienen IDs in OWL lediglich der Identifikation und besitzen in der Regel selbst keine Bedeutung.

*Assumption* kann zu völlig anderen Schlussfolgerungen führen, als unter einer abgeschlossenen Weltansicht, wie sie etwa in Datenbanken eingenommen wird.

Schlussfolgerungen können nicht nur für eine fertige Ontologie benutzt werden, um Aussagen über gegebene Fakten zu prüfen oder abzuleiten. Sie sind auch während der Entwicklung einer Ontologie von großem Wert, da mit ihrer Hilfe die Formalisierung von Wissen auf der Basis von bereits formalisiertem Wissen unterstützt werden kann, etwa durch die automatische Einordnung von neuen Konzepten in die bestehende Ontologie und die beständige Prüfung der Konsistenz der Ontologie (siehe auch STEVENS ET AL. 2001, S. 2).

#### Entwicklungswerkzeuge und Zugriffsmechanismen

Für die Entwicklung von Ontologien stehen integrierte Entwicklungsumgebungen zur Verfügung, z.B. OBO-Edit (<http://oboedit.org/>) und Protégé (<http://protege.stanford.edu/>). Protégé ermöglicht die Spezifikation von Ontologien in OWL und die Ausführung von Inferenzmaschinen. Die Online-Version WebProtégé (<http://protegewiki.stanford.edu/wiki/WebProtege>) bietet für die kollaborative Entwicklung von Ontologien zahlreiche Hilfestellungen. Auch für die Entwicklung von TEDDY wird Protégé eingesetzt.

Für vorhandene Ontologien existieren verschiedene Wege der Verwendung und des Zugriffs. Zum einen können Ontologiekonzepte verwendet werden, um Daten zu annotieren, z.B. mit Hilfe von RDF. Mit SPARQL können OWL-Ontologien und auf diesen basierende Wissensbasen abgefragt werden. Die SWRL (*Semantic Web Rule Language*, HORROCKS ET AL. 2004) erlaubt die Formulierung von Regeln zum Schlussfolgern mit OWL-Ontologien. Oft können Ontologien auch mittels bereitgestellter Webschnittstellen oder Webservices abgefragt werden. So bietet z.B. der OLS (*Ontology Lookup Service*, <http://www.ebi.ac.uk/ontology-lookup/>) eine zentrale Abfrage für eine Vielzahl an Ontologien.

Die stetig wachsende Zahl an Ontologien und die angestrebte Wiederverwendung und Interoperabilität von Ontologien macht es notwendig, dass die Ontologien im Internet verfügbar sind. Zahlreiche Portale stellen Ontologien bereit. Im Gebiet der Biomedizin sind besonders hervorzuheben: BioPortal (WHETZEL ET AL. 2011) und die OBO Foundry (*The Open Biological and Biomedical Ontologies Foundry*, SMITH ET AL. 2007). TEDDY ist in BioPortal verfügbar.

#### 3.3.3. Entwicklung von Ontologien

In diesem Abschnitt sollen einige Aspekte der Ontologieentwicklung diskutiert werden. Insbesondere wird dabei auf Methodologien, Prinzipien und Konventionen der kollaborativen Ontologieentwicklung eingegangen. Abschließend werden Probleme der Ontologierevision beleuchtet.

Informatische Ontologien sollten nicht nur logisch präzise und ästhetisch prägnant sein, sie müssen auch eine Übereinkunft unter ihren Benutzern darstellen. Ontologieentwicklung ist somit auch immer Standardisierung des Sprachgebrauchs in einem Wissensbereich:

„In building ontologies we are writing social contracts.“ (GRUBER ET AL. 1994, S. 10)

Um diese Übereinkunft zu erreichen, ist es sinnvoll, Ontologie kollaborativ zu entwickeln. Dafür ist es hilfreich, Konventionen festzulegen und bestimmten Prinzipien zu folgen (siehe unten).

Die notwendige Übereinkunft unter den Benutzern führt zu einer methodischen Parallele zwischen Untersuchungen im Gebiet der philosophischen Ontologie und der Entwicklung informatischer Ontologien: Ausgangspunkt in beiden Fällen sind sprachanalytische Untersuchungen des Gebrauchs von Wörtern und der durch diese repräsentierten Begriffe.

Weiterhin gibt es für einen bestimmten Wissensbereich nicht nur eine Ontologie. Vielmehr richtet sich die Adäquatheit einer Ontologie nach den intendierten Anwendungen, die Ontologieentwicklung muss also anwendungszentriert sein (vgl. [Abschnitt 3.2.2](#)). Die im Folgenden angeführten Methodologien beziehen sich deshalb auch stets auf intendierte Anwendungen.

## Methodologien

Die Entwicklung einer Ontologie ist ein Spezialfall der Wissensmodellierung im Knowledge Engineering und folgt deshalb im Wesentlichen dessen Methodologie (siehe [Abschnitt 3.1.3](#)). Eine explizit formulierte Methodologie der Ontologieentwicklung findet sich in der richtungsweisenden Anleitung von [NOY ET AL. 2001](#), das auch heute noch aktuelle „Ontology Development 101“. Eine alternative Methodologie stellt etwa [USCHOLD ET AL. 1996](#) dar. [CORCHO ET AL. 2003](#) gibt einen Überblick über verschiedene Methodologien der Ontologieentwicklung. Im Folgenden wird auf wichtige Punkte eingegangen.

Wie oben beschrieben, soll die Ontologieentwicklung anwendungszentriert sein. Als Ausgangspunkt für die Entwicklung einer Ontologie schlägt [USCHOLD ET AL. 1996](#) (S. 29ff) deshalb vor, motivierende Anwendungsszenarien zu formulieren und daraus potentielle Anfragen an die zu entwickelnde Ontologie abzuleiten. Anschließend wird aus den Anfragen eine informelle Menge an Konzepten extrahiert, die notwendig ist, die Anfragen und die Antworten zu repräsentieren. Der erste Schritt der Ontologieentwicklung bei [NOY ET AL. 2001](#) (S. 5) fordert analog die Festlegung des Wissensbereiches und der intendierten Anwendungen einer zu entwickelnden Ontologie mittels „competency questions“, also Fragen, deren Beantwortung mit Hilfe der Ontologie möglich sein soll. Anschließend an die Suche nach geeigneten existierenden Ontologien (zweiter Schritt) werden im dritten Schritt dementsprechend wichtige Konzepte aufgezählt ([NOY ET AL. 2001](#), S. 6).

Die restlichen Schritte der Ontologieentwicklung in [NOY ET AL. 2001](#) (S. 6ff) sind offensichtlich: schrittweise Definition von Konzepten und der Konzepthierarchie (Schritt 4), Festlegung von Eigenschaften für Konzepte und deren Beschränkungen (Schritte 5 und 6) und schließlich die Erzeugung von Instanzen (Schritt 7). Der letzte Schritt betrifft dabei eher die Verwendung der Ontologie. Die Bedeutung von [NOY ET AL. 2001](#) liegt jedoch eher in der Unterstützung bei wichtigen Designentscheidungen, etwa der Entscheidung, ob eine begriffliche Unterscheidung von Objekten im Wissensbereich durch ein neues Konzept oder eine neue Eigenschaft repräsentiert werden soll (S. 17f). Eine weitere wichtige Entscheidung betrifft die Abgrenzung zwischen Konzepten und Individuen (S. 18f). Für solche Entscheidungen werden jeweils prägnante Kriterien sowie illustrierende Beispiele angegeben und die Abhängigkeit der Entscheidung von der intendierten Anwendung dargelegt.

#### Prinzipien und Konventionen

Methodologien der Ontologieentwicklung sind bestimmten Prinzipien verpflichtet und folgen meist bestimmten Konventionen. In NOY ET AL. 2001 sind diese Prinzipien nur implizit in der Beschreibung des Entwicklungsprozesses und in den Entscheidungskriterien enthalten. Bereits GRUBER 1993b stellt Designprinzipien für die Entwicklung verteilter und wiederverwendbarer Ontologien auf. Andere Prinzipien der Ontologieentwicklung finden sich in etwa in CIMINO 1998. Auch Arbeiten, die eine Kritik an der wohl bekanntesten biologischen Ontologie GO (*Gene Ontology*, GENE ONTOLOGY CONSORTIUM 2001) üben, stellen entsprechende Kriterien für Ontologien auf (KUMAR ET AL. 2004, SMITH ET AL. 2003 und SMITH ET AL. 2004). Die OBO Foundry stellt Prinzipien und Konventionen auf, die bei der Entwicklung biomedizinischer Ontologien eingehalten werden sollen. Diese beziehen sich sowohl auf den logischen Inhalt und die Annotationen der Ontologien als auch auf den kollaborativen Entwicklungsprozess. Eine Sammlung von Prinzipien der OBO Foundry findet sich auf <http://www.obofoundry.org/wiki/index.php/Category:Principles>. Die Prinzipien sind mit einer ID versehen („FP\_“ gefolgt von einer dreistelligen Nummer). Im Folgenden wird auf wichtige Prinzipien und Konventionen der OBO Foundry eingegangen.

**Top-Level Ontologie:** Eine Domainontologie sollte eine Spezialisierung einer geeigneten Top-Level Ontologie sein. Dies erhöht die Interoperabilität zwischen Ontologien und deren Wiederverwendbarkeit. CEUSTERS ET AL. 2003 fordert entsprechend, dass eine Ontologie sich auf eine solide ontologische Theorie (im philosophischen Sinne) stützen sollte. Die OBO Foundry fordert explizit die Verwendung der BFO (*Basic Formal Ontology*, GRENON ET AL. 2004) als Top-Level Ontologie (FP\_014).

**Relationen:** Auch die verwendeten Relationen sollten für die Erhöhung der Interoperabilität standardisiert sein. So sollten, wenn möglich, Relationen aus der RO (*OBO Relations Ontology*, <http://code.google.com/p/obo-relations/>) verwendet werden (FP\_007).

**Inhalt:** Eine Ontologie sollte möglichst wenig inhaltliche Überschneidungen mit existierenden Ontologien haben (FP\_018). Dies vermeidet doppelten Entwicklungsaufwand. Stattdessen sollten, soweit möglich, bestehende Ontologien importiert werden. Auf jeden Fall sollte jede Ontologie einen klar umrissenen Inhalt besitzen (FP\_005).

**Identifikatoren:** Für die Interoperabilität ist es zudem erforderlich, dass die IDs einer Ontologie einen eigenen Namensraum besitzen. Dies kann durch die Verwendung von IRIs mit eigenem Namensraum als IDs erreicht werden (FP\_003). Zusätzlich sollten die IDs nach einer festgelegten Struktur aufgebaut sein (vgl. etwa die „OBO Foundry Identifier Policy“, <http://www.obofoundry.org/id-policy.shtml>). Darüber hinaus sollen die IDs an sich keine Bedeutung tragen.

**Dokumentation:** Für die Wiederverwendbarkeit sollte die Ontologie dokumentiert sein, z.B. durch eine Fachpublikation (FP\_008).

**Definitionen:** Wie bereits oben angeführt, sollte für die ästhetische Prägnanz jedes Konzept einer Ontologie mit einer natürlichsprachlichen Definition versehen sein (FP\_006), idealerweise in aristotelischer Form (FP\_013).

**Verfügbarkeit:** Die Ontologie sollte frei verfügbar sein (FP\_001) und in einem Standard-Dateiformat, wie etwa [OBO Format](#) oder [OWL](#) vorliegen (FP\_002).

**Kollaboration:** Die Ontologie soll in einer kollaborativen Art und Weise unter Beteiligung der Anwender entwickelt werden (FP\_010).

**Namenskonventionen:** Für die Konzeptbezeichner ist eine Namenskonvention festzulegen (FP\_012). Ein Beispiel für eine solche Namenskonvention findet sich in [SCHOBER ET AL. 2009](#).

### Ontologierevision

Ontologien werden in der Regel beständig weiterentwickelt. Diese Entwicklung sollte durch Versionierung nachvollziehbar gestaltet und dokumentiert werden (vgl. [OBO Foundry-Prinzip FP\\_004](#)). Insbesondere sollten alte Versionen einer Ontologie weiterhin zugreifbar sein, damit Daten, die auf ältere Versionen referenzieren, interpretierbar bleiben. Eine Konvention ist, überflüssige Konzeptbeschreibungen in einer Ontologie nicht einfach zu löschen, sondern lediglich als veraltet zu kennzeichnen, z.B. in [OWL](#) durch die Annotation `owl:deprecated`. Die IDs bleiben somit erhalten, d.h. sie sind persistent. Referenzen auf das gelöschte Konzept bleiben gültig. Ob und wann eine Konzeptbeschreibung sich soweit geändert hat, dass sie ein neues Konzept repräsentiert und somit eine neue ID bekommen muss, ist eine schwierige Frage (vgl. <http://www.obofoundry.org/wiki/index.php/Identifiers>).

Die Revision von Ontologien bringt erhebliche Probleme mit sich. Angenommen, Daten sind mit einer alten Ontologieversion annotiert. Was bedeutet diese Annotationen bezüglich der neuen Ontologieversion? Es kann sein, dass sich die Konzepte in der neuen Version stark verändert haben. In diesem Fall wäre es besser, Schlussfolgerungen über die Daten mit Hilfe der alten Version zu ziehen. Es kann jedoch auch sein, dass in der neuen Version Fehler beseitigt wurden, sodass es besser wäre, Schlussfolgerung mit Hilfe der neuen Version zu ziehen. Meist liegt wahrscheinlich eine Mischung aus beiden Fällen vor. Auch die Erhaltung alter IDs und die Vergabe neuer IDs bei erheblichen Veränderungen löst dieses Problem nicht, zumal es auch sehr subjektiv ist, was eine erhebliche Änderung darstellt. Eigentlich kann diese Frage nur in Bezug auf eine konkrete Verwendung der Ontologie beantwortet werden.

Es mag verwundern, dass sich trotz dieser Problematik kaum Arbeiten mit der Ontologierevision beschäftigen. Es gibt außer der klassischen Arbeit von [NEBEL 1990](#), die sich mit der Revision von [TBoxen](#) und den Auswirkungen auf die gesamte Wissensbasis und die möglichen Schlussfolgerungen befasst, nur wenige Ansätze, die Veränderungen einer Ontologie formal zu beschreiben, etwa [CIMINO 1996](#) und [TUTTLE ET AL. 1996](#) für den biomedizinischen Bereich.

Zum Abschluss dieses Abschnitts soll noch eine dynamische Sicht auf Ontologien gegeben werden: Eine sich verändernde Ontologie (als Spezifikation einer Konzeptualisierung) lässt

sich als dynamisches System auffassen, dessen Interpretation im Idealfall gegen die Konzeptualisierung des repräsentierten Wirklichkeitsausschnitts konvergiert. Die persistenten IDs repräsentieren dabei die Konzepte der Konzeptualisierung. Revisionen der Ontologie betreffen entweder die (natürlichsprachlichen) Konzeptbeschreibungen oder die (formalsprachliche) Axiomatisierung. Änderungen der ersten Art dienen der Verbesserung der ästhetischen Prägnanz, Änderungen der zweiten Art dienen der Verbesserung der logischen Präzision der Ontologie. Ontologieentwicklung kann dabei als evolutionärer Prozess gesehen werden, in welchem Ontologieversionen bezüglich der intendierten Verwendungen selektiert werden.



## Zusammenfassung

Dieses Kapitel hat sich der formalen Repräsentation von Wissen gewidmet ([Abschnitt 3.1](#)). Dabei stellte sich heraus, dass die logikbasierte Wissensrepräsentation ein Problem hat: Die Interpretation der Zeichen einer Interpretation kann nicht innerhalb des Wissensrepräsentationssystems erfolgen. Eine Lösung dieses *Symbol Grounding Problems* stellen informatische Ontologien dar. Die ursprünglich philosophische Ontologie als Untersuchung der Struktur der Wirklichkeit wurde in der Wissensrepräsentation neu gedeutet als formale Spezifikation der Bedeutung von Symbolen, welche eine Schnittstelle zwischen Wissensrepräsentationssystem und Benutzern bereitstellt. Um diese Funktion des *Symbol Groundings* erfüllen zu können, müssen informatische Ontologien logisch präzise und ästhetisch prägnant spezifiziert werden ([Abschnitt 3.2](#)). Mit der Realisierung informatischer Ontologien hat sich der letzte [Abschnitt 3.3](#) beschäftigt.

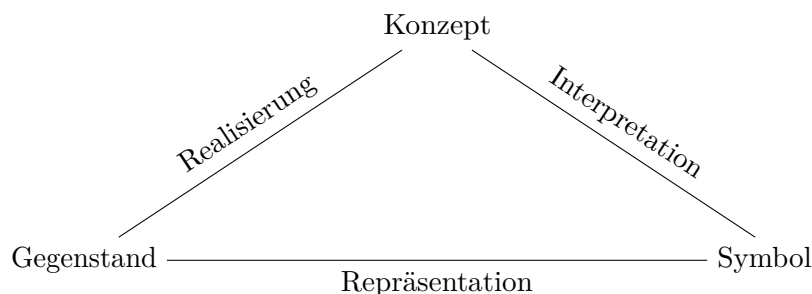


Abbildung 3.4.: Semiotisches Dreieck.

Die Schwerpunkte dieses Kapitels lassen sich im semiotischen Dreieck wiederfinden ([Abbildung 3.4](#)): Die philosophische Ontologie beschäftigt sich mit den Gegenständen in der Wirklichkeit, die Wissensrepräsentation stellt Wissen über diese Wirklichkeit symbolisch dar und informatische Ontologien erfassen die Bedeutung der durch die Symbole bezeichneten Konzepte. Allerdings erfolgt die Spezifikation von informatischen Ontologien wiederum durch Zeichen. Somit schließt sich ein weiteres semiotisches Dreieck an der Spitze von [Abbildung 3.4](#) an, in welchem die Symbole der Ontologiespezifikation die Konzepte der Wissensrepräsentation repräsentieren (siehe [SOWA 2000b](#), S. 60 für eine entsprechende Darstellung). Um den Symbolen dieser Ontologiespezifikation Bedeutung zu verleihen, müsste wiederum ein semiotisches Dreieck angesetzt werden und so weiter. Diesem unendlichen Regress kann nur entkommen werden, wenn einer informatischen Ontologie auf andere Weise eine Bedeutung verliehen wird: Einerseits durch ihre Verwendung in Schlussfolgerungen und andererseits durch ihren gemeinsamen Gebrauch auf der Grundlage des Agreements einer Gruppe menschlicher Benutzer. Dafür ist einerseits logische Präzision und andererseits ästhetische Prägnanz der Ontologie notwendig.

Dieses Kapitel erleichtert das Verständnis der formalen Repräsentation von Dynamik ([Kapitel 5](#)) und legt die theoretischen und technischen Grundlagen für die im Rahmen dieser Arbeit entwickelte Ontologie **TEDDY** ([Kapitel 6](#)) sowie für die im folgenden [Kapitel 4](#) skizzierte formale Repräsentation der Wissensfacetten von Bio-Modellen.



## 4. Formale Repräsentation der Wissensfacetten von Bio-Modellen

„However, there remains a distinct lack of standards for modeling or simulation.“

---

(KITANO 2002b: *Standards for modeling*, S. 337)

In [Kapitel 2](#) wurde mit den Wissensfacetten von Bio-Modellen ein konzeptuelles Schema entwickelt, welches eine systematische Beschreibung aller für das Modell und seiner Verwendung relevanten Wissensfragmente erlaubt. Dieses Kapitel beschreibt, wie das entsprechende Wissen formal repräsentiert werden kann, damit es die Basis für eine computergestützte Bio-Modellierung bilden kann, wie sie in [Abschnitt 1.3](#) vorgestellt wurde. In [Abschnitt 4.1](#) werden zunächst Bedingungen für die Computerunterstützung diskutiert. Bedingungen für die zumeist verteilt und kollaborativ erfolgende Bio-Modellierung sind insbesondere eine geeignete Infrastruktur und eine gewisse Standardisierung der bereitzustellenden Beschreibungen. Anschließend wird gezeigt, wie die einzelnen Wissensfacetten von Bio-Modellen adäquat formalisiert werden können ([Abschnitt 4.2](#)). Dabei wird, soweit möglich, auf existierende formale Mittel zurückgegriffen. Abschließend werden die entstandenen formalen Beschreibungen von Bio-Modellen in den wissenschaftlichen Kontext gestellt ([Abschnitt 4.3](#)). Erst dadurch kann die formale Repräsentation der Wissensfacetten von Bio-Modellen ihr volles Potential als Basis für eine computergestützte Modellierung und Simulation entfalten.

### 4.1. Bedingungen einer computergestützten Bio-Modellierung

Die computergestützte Bio-Modellierung bedarf einer Formalisierung der Wissensfacetten. Die verteilte und kollaborative Bio-Modellierung, die heutzutage der Regelfall ist (vgl. [Abschnitt 1.3.2](#)), stellt dabei besondere Anforderungen an die Computerunterstützung. Zunächst bedarf es einer gewissen Infrastruktur, um Bio-Modelle verteilt und kollaborativ zu entwickeln ([Abschnitt 4.1.1](#)). Für den Austausch und die Wiederverwendung von Bio-Modellen und den zugehörigen Daten muss sich als Erstes darüber geeinigt werden, welche Informationen jeweils zu einer vollständigen Beschreibung gehören. Mit solchen „Minimalen Informations-Standards“ beschäftigt sich der [Abschnitt 4.1.2](#). In [Abschnitt 4.1.3](#) wird untersucht, wie die Verbindung zwischen den intrinsischen und den extrinsischen Wissensfacetten erfasst werden kann.

#### 4.1.1. Infrastruktur für die Bio-Modellierung

Die Entwicklung und Verwendung von Modellen in der Systembiologie in der beschriebenen verteilten und kollaborativen Art und Weise (siehe [Abschnitt 1.3.2](#)) verlangt nach einer gut

ausgebauten Infrastruktur. Darunter fallen sowohl entsprechende Modellierungswerkzeuge (Editoren, Simulatoren, integrierte Entwicklungsumgebungen) als auch Austauschplattformen für Modelle (Datenbanken, Journale, Webseiten). Außerdem benötigt die Entwicklungscommunity eine geeignete Kommunikationsinfrastruktur (Webseiten, Mailinglisten, Entwicklergruppen, Entwicklertreffen). Darüber hinaus sind für die Verlinkung verteilt formalisierter Wissensfragmente bezüglich eines Bio-Modells besondere Mechanismen der Referenzierung notwendig, welche abschließend diskutiert werden.

Dabei soll dieser Abschnitt einen Überblick vermitteln, ohne dabei Anspruch auf eine vollständige Darstellung der existierenden Infrastruktur zu erheben. Die vorgestellten Werkzeuge und Projekte sind entweder sehr prominent oder wurden im Rahmen dieser Arbeit verwendet. Mit den eigentlichen Standards, formalen Sprachen und Ontologien, die für den Austausch und die Wiederverwendung von Modellen entwickelt werden, beschäftigen sich [Abschnitt 4.1.2](#) und [Abschnitt 4.2](#).

### Modellierungswerkzeuge

Für die Modellbildung und Modellmodifikation werden entsprechende Editoren benötigt. Die Bearbeitung eines Modells kann textbasiert oder grafisch erfolgen. Ein reiner textbasierter Editor für Bio-Modelle ist etwa der [SBMLeditor](#) ([RODRIGUEZ ET AL. 2007](#)), der direkte Manipulationen von [SBML](#)-Elementen erlaubt und dabei deren syntaktische Korrektheit überwacht. Gerade für die Bildung und Modifikation des durch das Modell spezifizierten Netzwerks bietet sich ein grafisches Vorgehen an, wie es z.B. durch den [CellDesigner](#) ([FUNAHASHI ET AL. 2003](#)) realisiert wird. Entsprechende Programme erlauben gleichzeitig auch eine Visualisierung eines Modells. Für die Festlegung des Programms eines Bio-Modells ([Facette S3](#)) wird meist zusätzlich zur grafischen Erstellung der Modellstruktur eine textbasierte Komponente benötigt. Eine kontextabhängige Unterstützung bei der Erstellung kinetischer Gleichungen bietet das Programm [SBMLsqueezer](#) ([DRAGER ET AL. 2008](#)).

Für die Kompetenz der Modelle ist es notwendig, die extrinsische Bedeutung der Modellkomponenten festzulegen. Häufig erfolgt dies über Annotationen der entsprechenden formalen Beschreibung. Eine Unterstützung bei der Annotation von Bio-Modellen erfolgt günstigerweise bereits während der Erstellung. [COPASI](#) (*C*OMplex *P*ATHWAY *S*IMULATOR, [HOOPS ET AL. 2006](#)) bietet diese Unterstützung. Daneben gibt es spezielle Programme, die auf die Aufgabe der Annotation spezialisiert sind, wie [Saint](#) ([LISTER ET AL. 2009](#)). In [semanticSBML](#) ([KRAUSE ET AL. 2010](#)) sind die extrinsischen Bedeutungen der zentrale Treiber für die Erstellung, Annotation, Visualisierung und Integration von Modellen.

Häufig werden für die Modellierung auch spezielle Programmiersprachen eingesetzt, wie etwa traditionell [Fortran](#) (<http://www.nag.co.uk/sc22wg5/>) und [MATLAB](#) (*MATrix LABoratory*, <http://www.mathworks.de/>) sowie zunehmend [R](#) (<http://www.r-project.org/>). Diese sind zwar dazu geeignet, dynamische Systeme zu spezifizieren sowie Simulationen und Analysen mit diesen durchzuführen. Jedoch ist bei diesen Sprachen die Modellstruktur nicht explizit gegeben und damit die Festlegung eines kompetenten Modells nicht möglich. Erschwerend kommt hinzu, dass bei der Benutzung von Programmiersprachen in der Modellierung oftmals die Modellstruktur und die Modellverwendung in dem Programmcode vermischt werden.

Insbesondere im *Systems Engineering* gibt es allgemeine Modellierungs- und Simulations-

werkzeuge, z.B. den **Wolfram SystemModeler** (<http://wolfram.com/system-modeler/>). Diese sind jedoch nicht auf Bio-Modelle zugeschnitten. Die Anpassung solcher Werkzeuge an die Bedürfnisse der Bio-Modellierung kann ein lohnendes Unterfangen sein, um von der langen Erfahrung der Modellierung im *Systems Engineering* zu profitieren.

Für die Simulation bzw. Analyse eines Bio-Modells werden entsprechende Simulatoren und Analyseprogramme benötigt. Häufig anzutreffen sind integrierte Entwicklungs- und Simulationsumgebungen, mit deren Hilfe die Modifikation und Verwendung von Modellen iteriert durchgeführt werden. Stellvertretend für solche integrierten Umgebungen sei **COPASI** genannt, mit der sich Modelle entwickeln und vielfältige Simulations- und Analyseaufgaben durchführen lassen. Die gewonnenen Resultate können grafisch dargestellt werden. Für aufwendige Simulationen (komplexe Modelle, stochastische Simulationen, Parameterscans) ist es jedoch vorteilhaft, den Simulationsalgorithmus als separates Programm verfügbar zu haben, um ihn in Stapelverarbeitung und in paralleler Verarbeitung (Cluster-Rechner) benutzen zu können. **COPASI** besitzt für die Stapelverarbeitung eine Kommandozeilenversion.

Neben der Integration verschiedener Aufgaben der Modellierung in einem Programm gibt es die Möglichkeit der Interaktion entsprechender Einzelprogramme. Dieses Vorgehen ist die Idee hinter der **SBW** (*Systems Biology Workbench*, [HUCKA ET AL. 2002](#)). Die **SBW** fungiert dabei als Kommunikationszentrale zwischen den einzelnen Programmen.

Für eine Übersicht von Modellierungswerkzeugen, die **SBML**-Modelle verarbeiten, siehe:

[http://sbml.org/SBML\\_Software\\_Guide/SBML\\_Software\\_Matrix](http://sbml.org/SBML_Software_Guide/SBML_Software_Matrix) und  
[http://sbml.org/SBML\\_Software\\_Guide/SBML\\_Software\\_Summary](http://sbml.org/SBML_Software_Guide/SBML_Software_Summary).

### Austauschplattformen

Für den Austausch von Modellen sind zunächst einmal entsprechende Originalpublikationen in Fachzeitschriften zuständig. Dort werden Modelle präsentiert, deren Verwendung beschrieben und die Simulationsergebnisse diskutiert. All dies erfolgt mit Bezug auf die biologischen Fragestellungen, die mit dem Modell beantwortet werden sollen.

Dieses Vorgehen hat drei wesentliche Nachteile, die zumindest eine Ergänzung durch andere Austauschwege notwendig machen:

1. Die natürlichsprachliche Darstellung ist für Computerprogramme nur schwer zugänglich. Eine direkte Überprüfung und Weiternutzung des präsentierten Modells in eigenen Simulationen ist somit nicht direkt möglich.
2. Große Modelle können in einer Publikation nicht vollständig dargestellt werden.
3. In Publikationen werden meist nur Modelle vorgestellt, die zu einer Beantwortung der biologischen Fragestellung einen Beitrag leisten. Modelle, die sich noch in der Entwicklung befinden und solche, die sich als ungeeignet für die vorliegende Aufgabe erwiesen haben, werden nicht auf diesem Wege ausgetauscht.

Es ist also zum einen notwendig, die formale Spezifikation der publizierten Modelle ergänzend zur Publikation verfügbar zu machen. Zum anderen ist ein Austausch von Modellen, die nicht in Fachzeitschriften publiziert sind, zu ermöglichen. Beide Ziele werden von entsprechenden Online-Datenbanken für Bio-Modelle verfolgt. Die wichtigste ist dabei die

BioModels Database (LE NOVÈRE ET AL. 2006), die in der 28. Release (September 2014) über 144 000 Bio-Modelle umfasst, von denen etwa 1200 in der Literatur veröffentlicht sind (die anderen sind automatisch generierte Modelle, für Details siehe [BIOMODELS DATABASE 2014](#)). Viele Projekte haben interne Modelldatenbanken von nicht veröffentlichten Modellen, z.B. [JWS Online](#) (*Java Web Simulation Online*, [SNOEP ET AL. 2002](#)) für das Projekt [Silicon Cell](#) ([SNOEP 2005](#)).

#### Infrastruktur für die Entwicklungscommunity

Die für verteilte und kollaborative Bio-Modellierung notwendigen Standards, formalen Sprachen und Werkzeuge müssen, um die notwendige breite Akzeptanz zu erlangen, von einer Community entwickelt werden, die einen engen Kontakt zu den späteren Benutzern pflegt. Im Fall der Bio-Modellierung besteht diese Community einerseits aus Informatikern, andererseits aus Systembiologen, die neben ihren eigentlichen Modellierungsprojekten die Entwicklung der Infrastruktur für die Bio-Modellierung vorantreiben.

Im Umfeld der Entwicklung von [SBML](#) (*Systems Biology Markup Language*, [HUCKA ET AL. 2003](#)) entstand eine solche gemischte Entwicklercommunity. Eine ähnliche Community sorgt sich um die Entwicklung von [CellML](#) ([CUELLAR ET AL. 2003](#)). Mit der Erkenntnis, dass für eine computergestützte Bio-Modellierung mehr Informationen über ein Modell formalisiert werden muss, als dessen Struktur, bildete sich mit [BioModels.net](#) ([LE NOVÈRE 2006a](#)) eine Community, die es sich zur Aufgabe machte, die notwendigen Standards, Ontologien und eine Modelldatenbank ([BioModels Database](#)) zu entwickeln.

Vor wenigen Jahren wurde erkannt, dass die parallelen Entwicklungen koordiniert werden müssen, um Redundanzen zu vermeiden, die Interoperabilität zu erhöhen und allgemein akzeptierte Standards zu etablieren. Mit [COMBINE](#) (*COmputational Modeling in BIOlogy NEtwork*, <http://co.mbine.org/>), ist eine Gemeinschaft von Entwicklercommunities entstanden, in der eine solche Koordinierung erfolgt. Diese Koordination erfolgt insbesondere durch zwei jährliche Treffen: Auf dem „COMBINE Forum“ werden unter Einbindung der Anwender neue Ideen entwickelt und bestehende Ansätze diskutiert. Der „HARMONY Hackathon“ dagegen dient der eigentlichen gemeinsamen Entwicklungsarbeit.

Für einen Kontakt zwischen Entwicklern und Nutzern werden moderne Kommunikationswege (Wikis, Mailinglisten, soziale Netzwerke) benutzt und entsprechende Meetings abgehalten. Durch das Feedback von den Nutzern soll eine Anpassung der Entwicklung an deren Bedürfnisse erfolgen.

Die von [COMBINE](#) koordinierten Standards und formalen Sprachen werden von demokratisch gewählten Editoren unter Beteiligung der Community entwickelt. Die entstehenden Spezifikationen sind frei zugänglich, die jeweiligen Kommunikationswege (Mailingliste) öffentlich. Dadurch soll eine erhöhte Akzeptanz der entwickelten Standards erreicht werden.

Oftmals gibt es ein streng definiertes Prozedere für die Entwicklungsprozesse eines Standards oder einer formalen Sprache. Die Entwicklungsprozesse von [SBML](#) sind wegen der Bedeutung dieser Sprache wohl mit am stärksten organisiert (siehe [http://sbml.org/Documents/SBML\\_Development\\_Process](http://sbml.org/Documents/SBML_Development_Process)).

Die Entwickler von Modellierungswerkzeugen, die eine formale Sprache wie etwa [SBML](#) durch das Programm verarbeiten lassen wollen, können auf verschiedene Weisen unterstützt werden: Der ausführlichen Spezifikation der Sprache wird eine entsprechende API (in Form einer Programmbibliothek) zur Seite gestellt, mit der Dateien in der formalen Sprache

gelesen, geschrieben und oft auch validiert werden können, z.B. [libSBML](#) ([BORNSTEIN ET AL. 2008](#)) für [SBML](#). Oft gibt es auch Testdateien, um die Konformität eines Werkzeugs bezüglich der Sprachspezifikation überprüfen zu können. So bietet die [SBML Test Suite](#) ([http://sbml.org/Software/SBML\\_Test\\_Suite](http://sbml.org/Software/SBML_Test_Suite)) die Möglichkeit, Simulatoren auf die Umsetzung bestimmter [SBML](#)-Eigenschaften hin zu überprüfen.

### Referenzierung externer Informationen

Die Repräsentation des Wissens über ein Bio-Modell erfolgt im Allgemeinen nicht innerhalb eines monolithischen Systems, sondern als verteilte Sammlung von Fragmenten in unterschiedlichen formalen Sprachen. Insbesondere die extrinsische Bedeutung wird meist durch Referenz auf Einträge in biologischen Datenbanken bestimmt. Die Umsetzung dieser Referenzen erfolgt durch Annotationen. Wie bereits in [Abschnitt 3.1.2](#) dargelegt, ist ein kritischer Aspekt die eindeutige Identifikation referenzierter Wissensfragmente. Dafür wurden in der Systembiologie spezielle Mechanismen entwickelt.

Die [MIRIAM Identifiers](#) ([JUTY ET AL. 2012](#)) stellen eindeutige und stabile IDs zur Verfügung. [MIRIAM Identifiers](#) sind als [URNs](#) (*Uniform Resource Name*) mit dem Präfix `urn:miriam` realisiert. Das Projekt [identifiers.org](#) ([JUTY ET AL. 2012](#)) stellt [URLs](#) (*Uniform Resource Locator*) für die [MIRIAM Identifiers](#) im Namensraum <http://identifiers.org/> bereit. Die Stabilität der IDs wird durch die [MIRIAM Registry](#) ([JUTY ET AL. 2012](#)) gewährleistet. Für dort registrierte Ressourcen erfolgt eine Auflösung der permanenten [MIRIAM Identifiers](#) in die aktuelle physikalische Adresse. Die Einträge in der [MIRIAM Registry](#) sind manuell erstellt und werden beständig aktualisiert. Im Folgenden werden, soweit wie möglich, ausschließlich [identifiers.org URLs](#) zur Referenzierung verwendet.

#### Beispiel 4.1 ([identifiers.org ID](#) für Modell 1)

Der Eintrag für das Modell 1 in der [BioModels Database](#) kann durch folgende URL referenziert werden:

<http://identifiers.org/biomodels.db/BIOMD0000000005>.

Dabei stellt <http://identifiers.org> den Präfix dar, [biomodels.db](#) identifiziert die Ressource und [BIOMD0000000005](#) ist eine interne ID innerhalb dieser Ressource (für Details siehe [JUTY ET AL. 2012](#)). Diese ID wird momentan (01.10.2014) zu folgender physikalischen Adresse aufgelöst: <http://www.ebi.ac.uk/biomodels-main/BIOMD0000000005>. ■

Die Referenzierung einer externen Information kann durch sogenannte Qualifikatoren genauer bestimmt werden. Die [BioModels.net Qualifiers](#) (<http://co.mbine.org/standards/qualifiers>) sind auf die Bedürfnisse von Bio-Modellen zugeschnitten. Die Benutzung der Qualifikatoren kann z.B. als Prädikat in [RDF](#) (*Resource Description Framework*, [MANOLA ET AL. 2004](#)) erfolgen.

Für [SBML](#) (Level 3) wird momentan mit dem [SBML Annotation Package](#) ([WALTEMATH ET AL. 2011c](#)) eine Erweiterung entwickelt, welche einen einheitlichen Mechanismus für die Annotation der Modellstruktur mit extrinsischer Bedeutung zur Verfügung stellt. Der Autor der vorliegenden Arbeit ist in diese Entwicklung involviert.



##### 4.1.2. Minimale Informations-Standards

Verteilte und kollaborative Bio-Modellierung setzt voraus, dass Beschreibungen von Modellen, Simulationsexperimenten und deren Resultate zwischen Wissenschaftlern ausgetauscht werden. Damit die ausgetauschten Beschreibungen nachvollziehbar, wiederholbar und wiederverwendbar sind, muss sich zunächst darüber verständigt werden, welche konkreten Informationen ausgetauscht werden sollen. Dies entspricht einem grundsätzlichen Anliegen der Wissenschaften: Ergebnisse einer Forschung sollen reproduzierbar sein, d.h. andere Wissenschaftler sollen nachvollziehen können, wie die Ergebnisse erreicht werden. Dafür muss den Ergebnissen eine detaillierte Beschreibung der notwendigen Voraussetzungen, der eingesetzten Mittel und Methoden beigelegt werden.

Da die für die Reproduzierbarkeit relevanten Informationen je nach Beschreibungsgegenstand variieren und Wissenschaftler verschiedene Auffassungen darüber haben, wann ein Ergebnis nachvollziehbar ist, wird sich häufig in einer wissenschaftlichen Community auf Beschreibungsstandards für die jeweiligen Untersuchungsmethoden geeinigt. Mit zunehmender Menge an ausgetauschten Beschreibungen und der Computer-Unterstützung bei der Verwertung solcher Beschreibungen wächst auch die Notwendigkeit der Standardisierung der Beschreibungen.

In den Lebenswissenschaften und insbesondere in der Systembiologie wurden in den letzten Jahren zahlreiche Richtlinien für spezielle wissenschaftliche Beschreibungen entwickelt, die als „reporting standards“ oder „minimum information guidelines“ ([TAYLOR ET AL. 2008](#)) bezeichnet werden. Der erste Name betont den normativen Aspekt („Standard“), der zweite Name bezieht sich auf den Umfang der Beschreibungen („Minimum“). Um beide Aspekte zu vereinen, soll im Folgenden der Name „MI-Standard“ (minimaler Informations-Standard) verwendet werden. Oft findet sich auch die Bezeichnung „Checkliste“ ([TAYLOR ET AL. 2008](#)), da die Standards oft in Gestalt einer Liste von bereitzustellenden Informationen auftreten.

Beschreibungen von Bio-Modellen, genauer der Modellstruktur, sollten dem [MI-Standard MIRIAM](#) (*Minimal Information Required In the Annotation of Models*, [LE NOVÈRE ET AL. 2005](#)) folgen. [MIRIAM](#) fordert dabei für die Beschreibung eines Bio-Modells (für Details siehe [LE NOVÈRE ET AL. 2005](#)):

1. die Implementierung der Modellstruktur in einer geeigneten formalen Sprache,
2. die Angabe der extrinsischen strukturellen Bedeutung durch entsprechende Annotationen,
3. die Angabe globaler Metainformationen über Herkunft und Versionsgeschichte des Modells sowie über zugehörige Publikationen und rechtliche Bedingungen der Modellnutzung (vgl. [Abschnitt 2.1.2](#)) und
4. die Reproduzierbarkeit der in der zugehörigen Publikation präsentierten Simulationsergebnisse.

Die genaue Beschreibung, wie ein Modell in Simulationsexperimenten verwendet werden soll, um vorliegende Simulationsergebnisse zu reproduzieren, sollte gemäß des [MI-Standards MIASE](#) (*Minimum Information About a Simulation Experiment*, [WALTEMATH ET AL. 2011a](#)) vorgenommen werden. In [MIASE](#) werden folgende funktionale Informationen über die Verwendung eines Bio-Modells gefordert:

1. Angaben zur vollständigen Instantiierung des Modells ([Facette F2](#)),
2. eine detaillierte Beschreibung der durchgeführten Simulation ([Facette F3](#)) und
3. eine Beschreibung, wie die finalen Simulationsergebnisse gewonnen werden (Postprozessierung, [Facette F3](#)).

Beide Standards sind Teil der Initiative [MIBBI](#) (*Minimum Information for Biological and Biomedical Investigations*, [LE NOVÈRE ET AL. 2005](#)), die als Dach für biologische und biomedizinische [MI-Standards](#) fungiert. Dort finden sich auch [MI-Standards](#) für die Beschreibung biologischer Experimente, also der extrinsischen Funktion. Beispiele dafür sind [MIAME](#) (*Minimum Information About a Microarray Experiment*, [BRAZMA ET AL. 2001](#)) und [MIFlowCyt](#) (*Minimum Information about a Flow Cytometry Experiment*, [LEE ET AL. 2008](#)).

Ein [MI-Standard](#) charakterisiert die relevanten Wissensfragmente, die eine entsprechende Beschreibung enthalten muss. Sie kann damit als Grundlage für die Entwicklung einer entsprechenden formalen Beschreibungssprache dienen. So kann [SBML](#) als Implementierung von [MIRIAM](#) und [SED-ML](#) (*Simulation Experiment Description Markup Language*, [WALTEMATH ET AL. 2011b](#)) als Implementierung von [MIASE](#) aufgefasst werden.

Die in dieser Arbeit vorgestellten Wissensfacetten ([Kapitel 2](#)) können ebenfalls als umfassende Richtlinie für eine systematische Beschreibung von Bio-Modellen aufgefasst werden. [MIRIAM](#) und [MIASE](#) können als Konkretisierung dieser Richtlinie bezüglich der Struktur bzw. der Funktion angesehen werden.

#### 4.1.3. Verbindung der intrinsischen und extrinsischen Wissensfacetten

Bevor sich im folgenden [Abschnitt 4.2](#) der Formalisierung der einzelnen Wissensfacetten zugewandt wird, soll in diesem Abschnitt diskutiert werden, wie die Beziehungen zwischen der intrinsischen und der extrinsischen Ebene erfasst werden können.

Die *Modeling Relation* (vgl. [Abschnitt 1.1.1](#)) stellt eine Kongruenz zwischen der intrinsischen und der extrinsischen Ebene eines formalen Modells dar (siehe [Abbildung 1.8](#)). Strukturell wird diese Kongruenz zwischen einem formalen System und einem Zielsystem im Begriff der Kompetenz (siehe [Abschnitt 1.1.3](#)) erfasst. Der Begriff der Performanz dagegen bezieht sich auf eine Ähnlichkeit im Verhalten des formalen Systems und des Zielsystems bei sinnvoller Verwendung des Modells unter plausiblen Rand- und Anfangsbedingungen. Performanz erfasst also eine verhaltensbezogene Kongruenz, die durch eine funktionale Kongruenz induziert ist. Dabei kommen Kompetenz und Performanz einem Bio-Modell nicht per se zu, sondern sind subjektive Zuschreibungen in einem bestimmten Verwendungskontext (vgl. [Abschnitt 1.1.1](#)). Es bedarf also der Angabe zusätzlicher Informationen, um einem Bio-Modell Kompetenz und Performanz zuzuschreiben und damit die intrinsischen Wissensfacetten mit ihren jeweiligen extrinsischen Wissensfacetten zu verbinden.

#### Zuschreibung von Kompetenz

Für die Zuschreibung von Kompetenz zu einem Bio-Modell ist es zunächst notwendig, ein biologisches Zielsystem zu identifizieren ([Facette S1](#)) und eine strukturerhaltende Abbildung vom formalen System auf dieses Zielsystem anzugeben. Diese Abbildung ist ein

Homomorphismus zwischen zwei Graphen: dem Netzwerk des formalen Systems, bestehend aus Entitäten ([Facette S2](#)) und Relationen ([Facette S3](#)), und dem Netzwerk von Objekten ([Facette S2](#)) und Interaktionen ([Facette S3](#)) des Zielsystems. Dabei werden die Entitäten des formalen Systems auf Objekte des Zielsystems so abgebildet, dass Relationen zwischen Entitäten durch Interaktionen zwischen zugehörigen Objekten widergespiegelt werden. Von großer Bedeutung für die Kompetenz eines Bio-Modells und damit für seine Fähigkeit, mechanistische Erklärungen zeitlicher Phänomene zu liefern, ist die Zuordnung zwischen dem dynamischen System und den Mechanismen des Zielsystems. Diese Zuordnung erfolgt vermittelt über den oben beschriebenen Homomorphismus zwischen formalem und biologischem Netzwerk (siehe auch [Beispiel 4.2](#)). Um Zuordnungen zwischen dem dynamischen System und den Relationen des Netzwerks spezifizieren zu können, muss das dynamische System über eine gewisse Kompositionalität verfügen: Es müssen Teile identifizierbar sein, die jeweils Abhängigkeiten ausschließlich zwischen solchen Attributen beschreiben, deren zugehörige Entitäten durch eine Relation verbunden sind. Prominente Ansätze für die Modellierung biologischer Systeme, wie etwa kinetische Modelle biochemischer Reaktionsnetzwerke und boolesche Modelle von Genregulationsnetzen (siehe [Abschnitt 1.2.2](#)) erfüllen dieses Kriterium der Kompositionalität.

#### **Beispiel 4.2 (Zuschreibung von Kompetenz zu Modell 1)**

In [Abbildung 4.1](#) ist ein Ausschnitt aus dem dynamischen System von **Modell 1** vermittelt über das Netzwerk des formalen Systems und das biochemische Reaktionsnetzwerk einem Mechanismus zugeordnet. ■

#### **Zuschreibung von Performanz**

Die Zuschreibung von Performanz zu einem Bio-Modell erfordert eine Bezugnahme auf biologische Experimente und die dabei beobachteten zeitlichen Phänomene. Zunächst muss dafür das Bio-Modell sinnvoll verwendet werden, um die biologischen Fragen ([Facette F1](#)) beantworten zu können. Mit dem Zusammenhang zwischen intendierter Verwendung eines Bio-Modells und den zu beantwortenden biologischen Fragen beschäftigt sich [Abschnitt 4.3.3](#). Die Verwendung des Modells muss unter plausiblen Bedingungen stattfinden, d.h. die Instantiierung und Initialisierung ([Facette F2](#)) des Modells muss in Einklang mit den Anfangs- und Randbedingungen ([Facette F2](#)) der zugehörigen biologischen Experimente erfolgen. Die Simulation muss in einer Art und Weise ([Facette F3](#)) durchgeführt werden, die den zugehörigen biologischen Experimenten angemessen ist. Performanz bedeutet die Fähigkeit der Imitation der am biologischen System beobachteten zeitlichen Phänomene. Der Vergleich zwischen Verhalten des Modells und dem beobachteten Verhalten des Zielsystems kann auf allen drei Unterfacetten des Verhaltens erfolgen: Es können sowohl die Simulationsergebnisse mit den Messergebnissen ([Facette B1](#)), die Ausgaben und Charakteristiken der Simulation mit den Darstellungen und Merkmalen der biologischen Experimente ([Facette B2](#)), als auch die Dynamiken mit den Phänomenen ([Facette B3](#)) verglichen werden.

## **4.2. Formalisierung der Wissensfacetten von Bio-Modellen**

In diesem Abschnitt wird gezeigt, wie sich die Wissensfacetten von Bio-Modellen formalisieren lassen. Dabei werden existierende Ansätze angeführt und Defizite diskutiert. Die

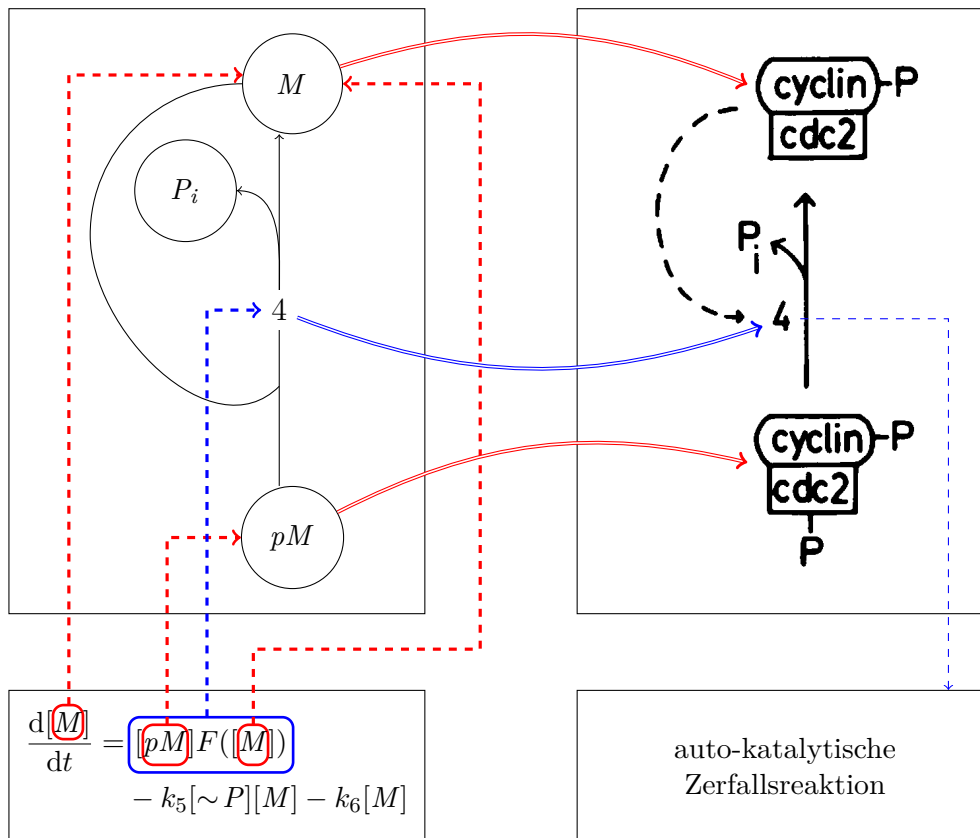


Abbildung 4.1.: Verbindung intrinsischer (links) und extrinsischer (rechts) Struktur von Modell 1. Zuordnung (gestrichelt) von Variablen (rot) und einem Term (blau) von Gleichung 1.9 (links unten) zu Entitäten und Relationen des formalen Systems (links oben, Ausschnitt aus Abbildung 2.3), Kongruenz (doppelte Linien) zwischen formalem System und biologischem Zielsystem (rechts oben, Ausschnitt aus TYSON 1991, Fig. 1, S. 7328, siehe auch Abbildung 1.3) sowie Zuordnung (gestrichelt) eines Mechanismus (rechts unten) zu einer Interaktion im Zielsystem.

Darstellung folgt dabei der Dreiteilung der Modellbedeutung in Struktur (Abschnitt 4.2.1), Funktion (Abschnitt 4.2.2) und Verhalten (Abschnitt 4.2.3). Zur Illustration werden die Wissensfacetten von Modell 1 formalisiert. In Tabelle 4.1 auf Seite 147 (ursprünglich veröffentlicht in KNÜPFER ET AL. 2013b) sind die erwähnten formalen Mittel bezüglich der Wissensfacetten zusammengefasst.

#### 4.2.1. Formalisierung der Modell-Struktur

##### Intrinsische Struktur

Die intrinsische Struktur eines Modells kann durch eine geeignete Modellierungssprache formal spezifiziert werden. Die wichtigsten Sprachen für die Struktur von Bio-Modellen sind SBML (*Systems Biology Markup Language*, HUCKA ET AL. 2003) und CellML (CUELLAR ET AL. 2003). Beide Sprachen sind XML-basiert. In SBML kann das Netzwerk eines

Bio-Modells explizit repräsentiert werden (siehe unten). Dabei werden die durch Relationen repräsentierten Reaktionen als eigenständig betrachtet, also eine prozessorientierte Sicht eingenommen. Dagegen ist [CellML](#) komponentenorientiert: Komponenten ([component](#)) sind dabei eigenständige Funktionseinheiten, die Variablen ([variable](#)) und mathematische Beziehungen ([math](#)) zwischen Variablen enthalten. Zwischen Komponenten erfolgt lediglich ein Austausch von Informationen über Zuordnungen ([connection](#)) von Variablen. Durch diese komponentenorientierte Sicht werden Informationen über [Facette S2](#) (Entitäten, Attribute) und [Facette S3](#) (Relationen, Programm) sowohl innerhalb einer Komponente vermischt, als auch zwischen Komponenten verteilt. Beides beeinträchtigt die Kompetenzfähigkeit des spezifizierten Bio-Modells. [CellML](#) ist jedoch gut geeignet, das dynamische System eines Bio-Modells zu spezifizieren.

Es existieren auch höhere Beschreibungssprachen für Modelle, wie etwa [Antimony](#) ([SMITH ET AL. 2009](#)), mit welchen sich Modelle einfacher spezifizieren lassen. In [Antimony](#) spezifizierte Modelle lassen sich in [SBML](#) semantikerhaltend konvertieren („kompilieren“). Die folgenden Ausführungen beziehen sich zumeist auf [SBML](#), da [SBML](#) einerseits die etabliertere Modellierungssprache ist und andererseits die strukturellen Facetten besser widerspiegelt als etwa [CellML](#).

Die Festlegung des Formalismus erfolgt durch die Wahl der verwendeten formalen Sprache. Der Kalkül, der dem *dynamischen System* ([Facette S1](#)) zugrunde liegt, ist meist implizit in der Implementierung des Modells bestimmt. Der Kalkül und seine Eigenschaften können explizit durch [SBO](#) (*Systems Biology Ontology*, [COURTOT ET AL. 2011](#)) bestimmt werden und zwar durch Konzepte aus dem Zweig 'modelling framework' ([SBO:0000004](#)). Der Zweig 'modeling and simulation algorithm characteristic' ([KISAO\\_0000097](#)) von [KiSAO](#) (*Kinetic Simulation Algorithm Ontology*, [COURTOT ET AL. 2011](#)) kann ebenfalls verwendet werden, um Eigenschaften des verwendeten Kalküls anzugeben.

Die *Entitäten* ([Facette S2](#)) und *Relationen* ([Facette S3](#)) werden durch die verwendete formale Sprache beschrieben. In [SBML](#) sind das die Elemente `listOfSpecies` für die Entitäten ([species](#)) und `listOfReactions` für die Relationen ([reaction](#)). Dabei besitzt jede [species](#) eine eindeutige ID, welche gleichzeitig als Variable (*Attribut*, [Facette S2](#)) im dynamischen System verwendet wird. Jeder [reaction](#) ist ein [kineticLaw](#) zugeordnet, welches das *Programm* ([Facette S3](#)) in Hinsicht auf diese Reaktion darstellt. Ein [kineticLaw](#) kann formale Parameter ([parameter](#)) enthalten. Das dynamische System wird durch die Menge all dieser [kineticLaws](#) festgelegt.

### Extrinsische Struktur

Die extrinsische Struktur wird einerseits durch ein Netzwerk aus *Objekten* ([Facette S2](#)) und *Interaktionen* ([Facette S3](#)) festgelegt und umfasst andererseits die biologische Bedeutung der Komponenten dieses Netzwerks.

Für bestimmte *epistemische Ebenen* des *biologischen Systems* ([Facette S1](#)) gibt es Standards und formale Sprachen zur Festlegung des Netzwerks. Für biochemische Reaktionsnetzwerke existiert zum Beispiel die formale Sprache [BioPAX](#) (*Biological Pathways Exchange*, [DEMIR ET AL. 2010](#)) zur Beschreibung von Stoffwechselwegen. Bei [BioPAX](#) handelt es sich um eine Ontologie, mit der Wissen über die statische Netzwerkstruktur biochemischer Systeme repräsentiert werden kann. Einen Standard für die grafische Repräsentation von biochemischen Systemen stellt [SBGN](#) (*Systems Biology Graphical*

*Notation*, [LE NOVÈRE ET AL. 2009](#)) dar. Von besonderem Interesse im Zusammenhang mit dynamischen Bio-Modellen ist die Untersprache **SBGN-PD** (**SBGN** Process Description Language, [MOODIE ET AL. 2009](#)), die sich der grafischen Darstellung von biochemischen Prozessen widmet. Ein **SBGN-PD**-Graph ist ein bipartiter Graph, in welchem eine Sorte von Knoten („entity pool nodes“) chemische Spezies repräsentieren und die andere Sorte Knoten („prozess nodes“) Prozesse, in welche die chemische Spezies involviert sind. In [Abbildung 4.2](#) ist das Reaktionsnetzwerk von **Modell 1** in **SBGN-PD** dargestellt. **SBGN** legt nur die Elemente (als „glyphs“ bezeichnet) der grafischen Darstellung und die erlaubten Verbindungen zwischen diesen fest. Für den Austausch zwischen Programmen gibt es eine entsprechende formale Sprache für **SBGN**, in welcher auch zusätzlich Layout-Informationen repräsentiert werden können.

Die extrinsische Interpretation der Struktur kann im Wesentlichen durch Annotationen auf entsprechende biologische Datenbanken und Ontologien formal gefasst werden. Auch **BioPAX** bietet die Möglichkeit, die biologische Bedeutung von Netzwerkkomponenten durch externe Verweise festzulegen.

*Biologische Systeme* ([Facette S1](#)) werden z.B. von der **GO** (*Gene Ontology*, [GENE ONTOLOGY CONSORTIUM 2001](#)) in der Teilontologie 'Biological Process' ([GO:0008150](#)) erfasst. Außerdem kann das biologische System mit Hilfe geeigneter biologischer Ressourcen näher spezifiziert werden, etwa durch Angabe der modellierten Organismen ([Taxonomy Database](#), [SAYERS ET AL. 2009](#)), des Entwicklungsstadiums (organismenspezifisch, etwa [XAO](#), *Xenopus Anatomy and Development Ontology* für Krallenfrösche) und des Zell- bzw. Gewebetyps ([CL](#), *Cell Ontology*, [BARD ET AL. 2005](#) und [PO](#), *Plant Ontology*, [COOPER ET AL. 2013](#)).

Es ist keine Ressource bekannt, welche die *epistemische Ebene* ([Facette S1](#)) explizit beschreibt, auf der biologische Systeme betrachtet werden. Meist ist die epistemische Ebene implizit durch die biologischen Untersuchungsmethoden bestimmt, die Grundlage für die Modellierung sind.

Zahlreiche biologische Datenbanken beinhalten Wissen über chemische Substanzen, wie etwa **UniProt** ([APWEILER ET AL. 2004](#)) für Proteine, **GenBank** ([BENSON ET AL. 2013](#)) für Gene, und **ChEBI** (*Chemical Entities of Biological Interest*, [DEGTYARENKO ET AL. 2008](#)) für Metaboliten und anorganische Verbindungen. Damit können die durch das Modell repräsentierten *Objekte* ([Facette S2](#)) bestimmt werden. Der **SBO**-Zweig 'material entity' ([SBO:0000240](#)) kann dazu verwendet werden, Objekte einer Substanzklasse zuzuordnen. Es gibt auch Substanz-Datenbanken, die auf einzelne Organismen spezialisiert sind, wie etwa **EcoCyc** ([KESELER ET AL. 2013](#)) für das Bakterium *Escherichia coli*.

Die *physikalischen Größen* ([Facette S2](#)), die durch die Attribute im Modell repräsentiert werden, können mit Hilfe von **SBO** als ein entsprechender 'systems description parameter' ([SBO:0000545](#)) charakterisiert werden. In den entsprechenden Konzepten sind bereits teilweise Angaben zu den Einheiten der physikalischen Größen hinterlegt. Eine umfassende Ontologie von physikalischen Größen stellt **PATO** (*Phenotypic Quality Ontology*, [GKOUTOS ET AL. 2004](#)) dar. Die momentan in Entwicklung befindliche **UO** (*Units Ontology*, [GKOUTOS ET AL. 2012](#)) beinhaltet physikalische Einheiten und verlinkt diese mit entsprechenden Einträgen in **PATO** via der Relation `is_unit_of`.

Für biochemische Reaktionen (*Interaktionen*, [Facette S3](#)) gibt es zahlreiche Datenbanken, von welchen hier nur **Reactome** ([JOSHI-TOPE ET AL. 2005](#)) genannt werden soll. Die „EC Numbers“ („EC“ steht historisch für „Enzyme Commission“, welche aktuell von der **NC-**



IUBMB (*International Union of Biochemistry and Molecular Biology*, <http://www.iubmb.org/>) verwaltet werden, sind eine Klassifikation enzymatischer Reaktionen und finden sich in der Datenbank *IntEnz* (*Integrated relational Enzyme database*, FLEISCHMANN ET AL. 2004). Die *Mechanismen* (Facette S3), die die Dynamik der Reaktionen beschreiben, können durch SBO-Konzepte unter 'rate law' (SBO:0000001) spezifiziert werden. Die entsprechenden SBO-Einträge enthalten auch die mathematischen Gleichungen der Geschwindigkeitsgesetze.

### Verbindung intrinsischer und extrinsischer Struktur

Die meisten Modellierungssprachen bieten die Möglichkeit, Modellkomponenten mit Annotationen zu versehen und somit eine extrinsische Bedeutung zuzuweisen. In SBML ist dies über das Element `annotation` realisiert, welches einen RDF-Ausdruck enthalten darf. In diesem können *BioModels.net Qualifiers* benutzt werden, um die Beziehung (Prädikat) zwischen der umgebenden Modellkomponente (Subjekt) und dem Annotationsziel (Objekt) zu spezifizieren. Mit Hilfe geeigneter Annotationen kann in SBML dem intrinsischen Netzwerk eines Modells ein korrespondierendes extrinsisches Reaktionsnetzwerk zugeordnet werden.

#### Beispiel 4.3 (Formalisierung der Struktur von Modell 1)

In diesem Beispiel wird die Formalisierung der strukturellen Bedeutung von Modell 1 diskutiert, welche in Beispiel 2.3 dargestellt wurde. Im Folgenden werden Ausschnitte aus der SBML-Codierung von Modell 1 (aus dem Eintrag BIOMD0000000005 der *BioModels Database*) gezeigt. Dabei werden teilweise nicht relevante XML-Attribute und -Elemente weggelassen. Die vollständige SBML-Datei von Modell 1 findet sich in Anhang C.

Für die intrinsische Struktur muss zunächst der benutzte Kalkül des *dynamischen Systems* (Facette S1) festgelegt werden: Dies kann für Modell 1 mit Hilfe von SBO als 'non-spatial continuous Framework' (SBO:0000293) erfolgen. Das *formale System* (Facette S1) von Modell 1 ist durch die `listOfSpecies` und `listOfReactions` in SBML implementiert. Dabei werden die *Entitäten* (Facette S2) durch `species`-Elemente und die *Relationen* (Facette S3) durch `reaction`-Elemente spezifiziert. Die `kineticLaw`-Elemente legen zusammen das *Programm* (Facette S3) fest. Z.B. beschreiben folgende Ausschnitte aus der SBML-Datei von Modell 1 (Anhang C, Zeilennummern wie dort) die Entität C2 sowie die Relation von M zu C2 und YP mit zugehörigem Programm:

```

154 <species id="C2" initialAmount="0" name="cdc2k" metaid="_000004" compartment="
    ↪ cell">
169 </species>
310 <reaction id="Reaction1" name="cyclin_cdc2k dissociation" metaid="_000010"
    ↪ reversible="false">
330   <listOfReactants>
331     <speciesReference species="M" metaid="_712369"/>
332   </listOfReactants>
333   <listOfProducts>
334     <speciesReference species="C2" metaid="_712382"/>
335     <speciesReference species="YP" metaid="_712394"/>
336   </listOfProducts>
337   <kineticLaw metaid="_712406">
338     <math xmlns="http://www.w3.org/1998/Math/MathML">
339       <apply>
340         <times/>
341         <ci> cell </ci>

```



```

342     <ci> k6 </ci>
343     <ci> M </ci>
344   </apply>
345 </math>
346 <listOfParameters>
347   <parameter id="k6" metaid="_840210" value="1"/>
348 </listOfParameters>
349 </kineticLaw>
350 </reaction>

```

Die *metaid* dient dabei als Subjekt-Referenz für Annotationen in [RDF](#) (siehe unten). Zu beachten ist, dass in dieser [SBML](#)-Implementierung die Reaktionen anders nummeriert sind. So bezeichnet *Reaction1* im gegebenen [SBML](#)-Fragment die Relation 6 im formalen System von Modell 1 gemäß [Abbildung 2.3](#).

Die *Attribute* ([Facette S2](#)) sind implizit durch [SBML](#) festgelegt: Zu jeder Entität (*species*) gehört eine Variable (definiert durch das *id*-Attribut), die die Konzentration dieser Entität repräsentiert. Die Variablen werden innerhalb der *kineticLaw* verwendet. Ein *kineticLaw* kann *parameter* beinhalten. Im angegebenen Beispiel ist das  $k_6$ .

Die extrinsische Struktur von Modell 1 beschreibt das modellierte *biologische System* ([Facette S1](#)). Das Reaktionsnetzwerk von Modell 1 (vgl. [Abbildung 1.3](#)) kann mit Hilfe der [SBGN-PD](#) formal repräsentiert werden, siehe [Abbildung 4.2](#).

Für Modell 1 kann das biologische System durch GO als 'mitotic cell cycle' ([GO:0000278](#)) bestimmt werden und der adressierte Organismus z.B. durch die [Taxonomy Database](#) als Seeigel 'Echinoidea' (ID [7625](#)). Um alle von [TYSON 1991](#) adressierten Organismen zu erfassen, könnte auch eine höhere taxonomische Einheit angegeben werden, die Seeigel und Frösche umfasst, etwa Neumünder 'Deuterostomia' (ID [33511](#)) oder noch genereller Gewebetiere 'Eumetazoa' (ID [6072](#)). Die höchste taxonomische Einheit, die für die Charakterisierung des biologischen Systems von Modell 1 noch sinnvoll erscheint, sind die vielzelligen Tiere 'Metazoa' (ID [33208](#)), da die von [TYSON 1991](#) beschriebenen embryonalen Prozesse nur in dieser Organismengruppe vorkommen.

Die durch die Modell-Entitäten repräsentierten biologischen *Objekte* ([Facette S2](#)) können durch entsprechende Annotationen charakterisiert werden. Z.B. repräsentiert *C2* die 'Cyclin-dependent kinase 1' ([UniProt: P04551](#)). In [SBML](#) dient für die Spezifikation dieser Annotation das entsprechende Unterelement der *species* (Zeilennummern entsprechend [Anhang C](#)):

```

155 <annotation>
156   <rdf:RDF
157     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
158     xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
159     xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
160     <rdf:Description rdf:about="#_000004">
161       <bqbiol:isVersionOf>
162         <rdf:Bag>
163           <rdf:li rdf:resource="http://identifiers.org/uniprot/P04551"/>
164         </rdf:Bag>
165       </bqbiol:isVersionOf>
166     </rdf:Description>
167   </rdf:RDF>
168 </annotation>

```

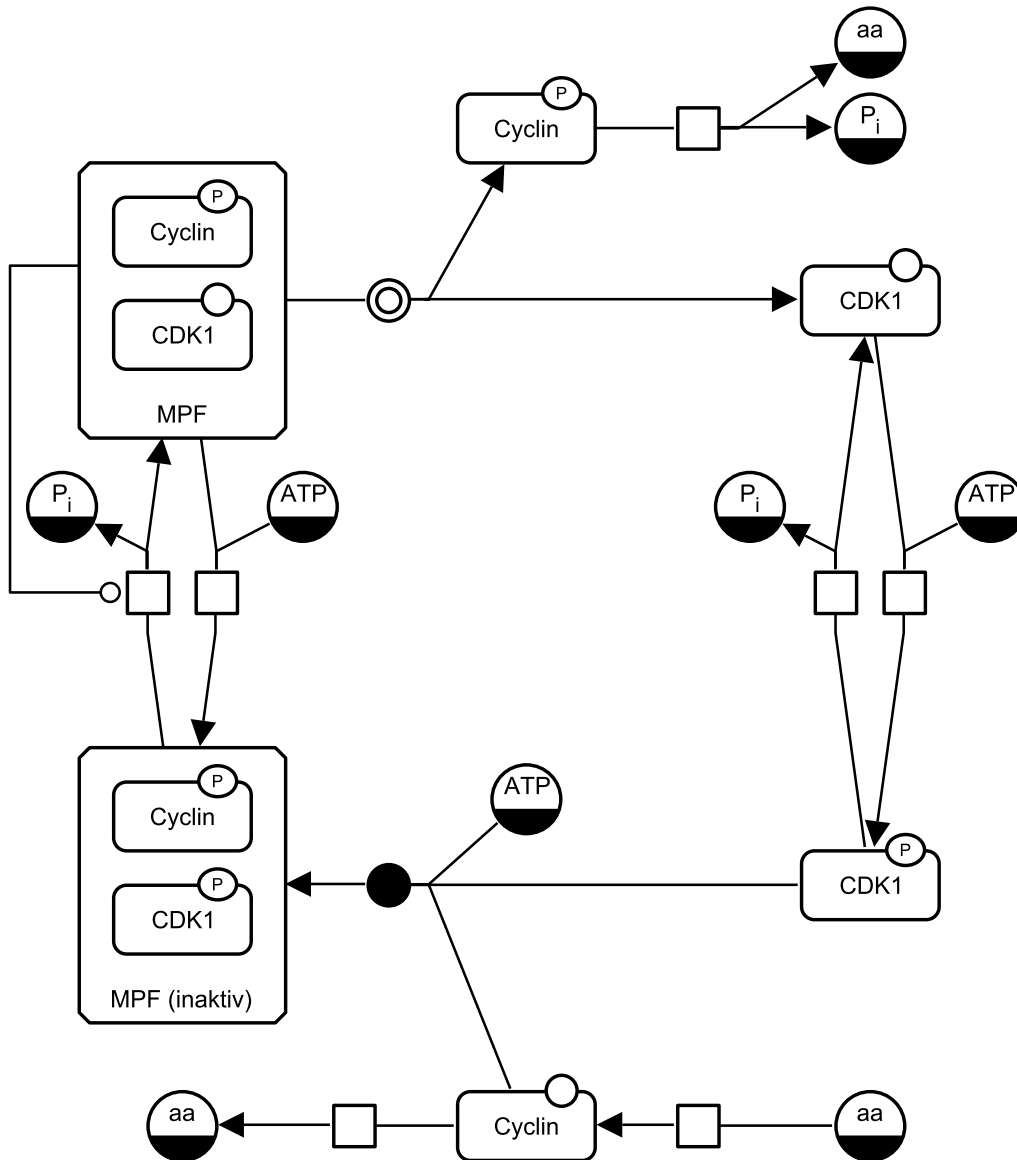


Abbildung 4.2.: Repräsentation des Reaktionsnetzwerks von Modell 1 (vgl. [Abbildung 1.3](#)) als **SBGN**-Graph (in der Untersprache **SBGN-PD**): Proteine sind repräsentiert durch abgerundete Rechtecke, die eine Kennzeichnung über den Phosphorylierungszustand enthalten können. Die Achtecke kennzeichnen die beiden Dimere aus **CDK1** und Cyclin. Andere chemische Spezies sind durch Kreise repräsentiert. Für eine bessere Übersichtlichkeit der Grafik treten einige chemische Spezies mehrfach auf, diese sind durch eine dunkle Unterseite im Kreis gekennzeichnet (in **SBGN** als „clone marker“ bezeichnet). Die Reaktionen sind durch gefüllte Kreise (Dimerisation), Doppelkreise (Dissoziation) bzw. Vierecke (alle anderen Reaktionen) dargestellt. Ungerichtete Kanten verbinden Reaktanten mit den Reaktionen, an denen sie teilnehmen. Gerichtete Kanten verbinden Reaktionen mit ihren Produkten. Die Kante mit dem Kreis an einem Ende repräsentiert den auto-katalytischen Einfluss von **MPF** auf die Reaktion 4. Der Graph wurde mit Hilfe von **SBGN-ED** erstellt, einem Add-on für **VANTED**.

Das Subjekt dieser Annotation ist durch die `metaid` bestimmt. Die Darstellung des RDF-Objekts als Menge (`rdf:Bag`) folgt dabei dem durch die SBML-Spezifikation empfohlenen Schema (HUCKA ET AL. 2008, S. 92ff).

Alle Objekte des Modells haben als *physikalische Größe* (Facette S2) ihre molare Konzentration (in mol/l) zugeordnet, in SBO 'molar concentration of an entity' (SBO:0000472). Die physikalische Größe kann alternativ auch durch PATO als 'concentration of' (PATO:0000033) erfolgen, mit der entsprechenden Einheit 'unit of molarity' (UO:0000061) aus der UO.

Die *Interaktion* (Facette S3), die durch die `Reaction1` in der SBML-Implementierung von Modell 1 repräsentiert wird, kann als 'Formation of Cyclin A:Cdc2 complexes' (Reactome: REACT\_6308) bestimmt werden. Genereller kann die Reaktion als Variante von 'regulation of cyclin-dependent protein serine/threonine kinase activity' (GO:0000079) betrachtet werden.

Der *Mechanismus* (Facette S3) der meisten Reaktionen von Modell 1 folgt dem 'mass action rate law' (SBO:0000012). Nur `reaction9` (Relation 4 in Abbildung 2.3) realisiert einen sehr speziellen autokatalytischen Feedback (vgl. TYSON 1991), für welchen kein spezifischer SBO-Term existiert.

SBO kann auch für eine grobe Klassifikation der Substanzen und Reaktionen verwendet werden, z.B. `C2` als 'polypeptide chain' (SBO:0000252) und `Reaction1` als 'dissociation' (SBO:0000180).

Modell 1 kann auch in CellML spezifiziert werden und findet sich etwa in der CellML Model Repository (LLOYD ET AL. 2008)<sup>1</sup>. In der BioModels Database ist eine BioPAX-Spezifikation von Modell 1 verfügbar<sup>2</sup>. ■

## 4.2.2. Formalisierung der Modell-Funktion

### Intrinsische Funktion

Eine formale Beschreibung der intrinsischen Funktion eines Bio-Modells entspricht einer vollständigen Spezifikation der Simulationsexperimente, die mit dem Modell durchgeführt werden. Eine solche Spezifikation von Simulationsexperimenten kann mit der formalen Sprache SED-ML erfolgen. SED-ML befindet sich zur Zeit in aktiver Entwicklung. Die aktuell vorliegende Version von SED-ML (Level 1 Version 2 vom 02.12.2013) erlaubt die Beschreibung sehr einfacher Simulationsexperimente und deren wiederholter Ausführung unter variierenden Bedingungen (siehe unten). Die meisten Simulatoren besitzen ihre eigene proprietäre Beschreibungssprache für Simulationen. Im Folgenden soll sich aber auf das standardisierte Dateiformat SED-ML konzentriert werden.

Die *intendierte Verwendung* (Facette F1) eines Bio-Modells wird in SED-ML durch die Elemente von `listOfSimulations` bestimmt. Neben dem `uniformTimeCourse`, also der Simulation einer Zeitreihe mit äquidistanten diskreten Zeitpunkten, gibt es den Simulationstyp `SteadyState` zur Berechnung stationärer Zustände. Für die Spezifikation von kombinierten Simulationen, wie etwa die Erstellung von Bifurkationsdiagrammen oder Parameteroptimierungen, wurde eine Erweiterung von SED-ML vorgeschlagen BERGMANN 2012. Durch die Wiederholung und Verschachtelung („nested simulation experiments“)

<sup>1</sup>[http://models.cellml.org/exposure/9bfff394be3ade829feed94151b3d68b3/tyson\\_1991.cellml/view](http://models.cellml.org/exposure/9bfff394be3ade829feed94151b3d68b3/tyson_1991.cellml/view)

<sup>2</sup><http://www.ebi.ac.uk/biomodels/models-main/publ/BIOMD0000000005/BIOMD0000000005-biopax3.owl>

einfacher Simulationen lassen sich demnach nahezu beliebige kombinierte Simulationen re-präsentieren. Einer möglichst hohen Ausdrucksfähigkeit der so definierten formalen Sprache steht die Notwendigkeit gegenüber, eine entsprechende Sprachunterstützung in Simulatoren zu implementieren. Die Abwägung der Bedürfnisse der Modellierer gegen die Herausforderungen für die Programmierer der Simulatoren ist Gegenstand einer längeren Kontroverse (siehe die Mailingliste [SED-ML-discuss](#)). Eine Umsetzung der Idee von [BERGMANN 2012](#) erfolgte mit dem Konstrukt `RepeatedTask` in der aktuellen Version 2 von [SED-ML Level 1](#). Dadurch können Simulationen unter variierenden Instantiierungen und Initialisierungen ([Facette F2](#)) wiederholt werden.

Die Formalisierung von *Constraints* ([Facette F1](#)), die bei der Verwendung des Modells eingehalten werden müssen, ist ein offenes Problem. Zwar existiert ein entsprechendes `Constraint`-Element in [SBML](#). Dessen Interpretation ist jedoch durch die [SBML](#)-Spezifikation ([HUCKA ET AL. 2008](#)) nicht vollständig festgelegt und wird aktuell diskutiert (z.B. auf der Mailingliste [sbml-discuss](#)). Die Verwendung von Ansätzen aus dem [CLP](#) (*Constraint Logic Programming*) scheint für die Spezifikation von Constraints vielversprechend. Allerdings müssten dann auch die entsprechenden *Constraint Solver* in die Simulatoren integriert werden.

Die *Instantiierung* und *Initialisierung* eines Bio-Modells ([Facette F2](#)) kann bereits bei der Festlegung der Modellstruktur in [SBML](#) vorgenommen werden. Für Parameter (`parameter`) existiert das entsprechende [XML](#)-Attribut `value`, über welches eine Instantiierung vorgenommen werden kann. Für die Initialisierung der Attribute existieren für jede Entität (`species`) die [XML](#)-Attribute `initialAmount` und `initialConcentration`, mit welchen die zugehörigen Variablen initialisiert werden können. Eine solche Festlegung der Modell-Instantiierung zusammen mit der Modell-Struktur ist aus zwei Gründen problematisch: Zum einen ist eine klare konzeptuelle Trennung von Modellstruktur und Modellverwendung für die Kommunikation von Modellen sehr hilfreich. Zum anderen wird ein Modell in der Regel das Zielsystem unter verschiedenen Randbedingungen und in unterschiedlichen Zuständen abbilden. Dafür sind unterschiedliche Instantiierungen des Modells notwendig. In [SED-ML](#) existiert für die Festlegung und Änderung der Parameterinstantiierung und der Attributinitialisierung das [XML](#)-Element `changeAttribute`.

Das *Setup* eines Simulationsexperiments sowie die *Postprozessierung* ([Facette F3](#)) der anfallenden Rohdaten wird in [SED-ML](#) spezifiziert. Dafür kann im [XML](#)-Element `algorithm` der zu verwendende Algorithmus für die entsprechende Simulation (`simulation`) durch das Attribut `kisaoID` bestimmt werden. Die `kisaoID` verweist auf die Ontologie [KiSAO](#), welche Simulationsalgorithmen beschreibt und klassifiziert. Das Setup eines Simulationsexperiments wird in [SED-ML](#) durch [XML](#)-Attribute von `simulation` spezifiziert. So kann z.B. über die beiden [XML](#)-Attribute `initialTime` und `numberOfPoints` der Startpunkt und die Anzahl von Zeitschritten festgelegt werden. Die Zuordnung zwischen Modellen und den Simulationen, die mit diesen durchgeführt werden sollen, erfolgt in [SED-ML](#) über das [XML](#)-Element `task`. Somit kann dieselbe Simulation auf mehrere Modelle bzw. Varianten desselben Modells angewendet werden. Andererseits können verschiedene Simulationen mit einem Modell durchgeführt werden. Die Nachbearbeitung der Rohdaten erfolgt in [SED-ML](#) über `dataGenerator`-Elemente. Die eigentliche Ausgabe der Simulation ([Facette B2](#)) wird durch die `listOfOutputs` festgelegt.

### Extrinsische Funktion

Zur Zeit existiert kein Ansatz, die *biologischen Fragen* (Facette F1), die durch die Verwendung eines Bio-Modells beantwortet werden sollen, formal zu fassen. In Abschnitt 4.3.2 wird eine Ontologie entworfen, die helfen kann, diesen Missstand zu beseitigen. Diese Ontologie klassifiziert Ziele der Modellverwendung und verbindet diese mit geeigneten Verwendungsweisen.

Für die Formalisierung der *Annahmen* (Facette F1), die den Constraints bei der Modellverwendung zugrunde liegen, ist zur Zeit ebenfalls kein Ansatz bekannt.

Die *Randbedingungen* und *initialen Zustände* (Facette F2), die durch die Modellverwendung nachgebildet werden sollen, können durch Annotationen der zugehörigen Parameter und initialer Werte für die Attribute festgelegt werden. Für die in der Datenbank SABIO-RK (*System for the Analysis of Biochemical Pathways – Reaction Kinetics*, WITTIG ET AL. 2012) gespeicherten kinetischen Daten sind die experimentellen Bedingungen ihrer Messung abgelegt. Eine qualitative Charakterisierung solcher Randbedingungen und initialer Zustände kann durch eine nähere Bestimmung des modellierten biologischen Systems mit Hilfe von Ontologien erfolgen. So kann z.B. der Zelltyp für Tiere durch die CL (*Cell Ontology*, BARD ET AL. 2005) und für Pflanzen durch die PO (*Plant Ontology*, COOPER ET AL. 2013) bestimmt werden. Viele der entsprechenden Ontologien sind jedoch oft auf bestimmte Organismen oder Organismengruppen spezialisiert, wie etwa die XAO (*Xenopus Anatomy and Development Ontology*) für die Entwicklungsstadien von Krallenfröschen. Andererseits existieren Ontologien, mit denen Bedingungen und Messmethoden biologischer Untersuchungen angegeben werden können. Diese sind jedoch auf spezielle Untersuchungsverfahren zugeschnitten, wie etwa die XCO (*Experimental Conditions Ontology*, SHIMOYAMA ET AL. 2012) für morphologische oder physiologische Untersuchungen und die MMO (*Measurement Method Ontology*, SHIMOYAMA ET AL. 2012) für klinische Untersuchungen.

Ähnlich wie bei der Bestimmung der experimentellen Bedingungen gibt es für die Spezifikation des *experimentellen Settings* und der *Auswertung* (Facette F3) auf bestimmte biologische Untersuchungsverfahren zugeschnittene formale Sprachen. Als Beispiel sei FuGE (*Functional Genomics Experiment Object Model*, JONES ET AL. 2007) für Experimente der funktionalen Genomik genannt. Da zahlreiche Untersuchungsverfahren existieren, mit denen die modellierten Phänomene beobachtet werden, soll hier nicht weiter auf die Formalisierung der extrinsischen Facette F3 eingegangen werden. Es ist festzuhalten, dass entsprechende Formalisierungsansätze zumindest für bestimmte biologische Untersuchungsmethoden verfügbar sind.

#### Beispiel 4.4 (Formalisierung der Funktion von Modell 1)

In diesem Beispiel soll gezeigt werden, wie die funktionalen Aspekte der in Beispiel 2.5 beschriebenen Modellverwendung formal spezifiziert werden können. Dabei geht es um die Reproduktion der Zeitreihe aus Fig. 3a in TYSON 1991 (vgl. Abbildung 1.5).

Die im Folgenden diskutierte Formalisierung der intrinsischen Funktion dieses Beispiels basiert auf einer Codierung in SED-ML, welche in Anhang E vollständig zu finden ist. Diese Codierung wurde mit Hilfe der SED-ML Web Tools ([http://sysbioapps.dyndns.org/SED-ML\\_Web\\_Tools/](http://sysbioapps.dyndns.org/SED-ML_Web_Tools/)) erstellt, mit welchen auch Abbildung 1.5 generiert wurde. In den folgenden Ausschnitten der SED-ML-Datei sind unwichtige XML-Elemente und XML-Attribute weggelassen.

Die *intendierte Verwendung* (Facette F1) von Modell 1 wird als normale Zeitreihe festgelegt (Zeilennummern entsprechend Anhang E):

```

4 <listOfSimulations>
5   <uniformTimeCourse id="sim1" initialTime="0"
6                       outputStartTime="0" outputEndTime="100"
7                       numberOfPoints="1000">
8     <algorithm kisaoID="KISA0:0000019" />
9   </uniformTimeCourse>
10 </listOfSimulations>

```

Damit wird zugleich das *Setup* (Facette F3) für die Simulation angegeben: Es soll eine Zeitreihe vom Zeitpunkt  $t = 0$  (outputStartTime) bis  $t = 100$  (outputEndTime) mit 1000 Zeitpunkten (numberOfPoints) berechnet werden. Zur numerischen Integration wird dabei der Algorithmus (algorithm) 'CVODE' (KISA0\_0000019) verwendet. TYSON 1991 verwendet ursprünglich die 'Adams-Moulton method' (KISA0\_0000280) für die Integration. CVODE (HINDMARSH ET AL. 2005) ist in der frei verfügbaren Programmbibliothek SUNDIALS (HINDMARSH ET AL. 2005) enthalten und ist in verschiedenen Simulatoren implementiert, unter anderem in den SED-ML Web Tools. Außerdem greift CVODE intern auf die Adams-Moulton-Methode zurück (HINDMARSH ET AL. 2005). Aus diesen Gründen wird hier CVODE verwendet. In COPASI wird mit LSODA (Livermore Solver for Ordinary Differential Equations with Automatic Method Selection, HINDMARSH 1983) eine sehr ähnliche Methode zur numerischen Integration verwendet (vgl. Beispiel 2.5), welche ebenfalls auf die Adams-Moulton-Methode zurückgreift. Die entsprechende kisaoID für 'LSODA' ist KISA0\_0000088.

Das *Constraint* (Facette F1)  $[CT] = \text{const.}$  ist bereits als boundaryCondition in der SBML-Implementierung festgelegt (Zeilennummer 268 in Anhang C):

```

268 <species id="CT" name="total_cdc2" metaid="metaid_0000022" boundaryCondition="
    ↪ true" compartment="cell">

```

Das Constraint  $k_2 \ll k_3[CT]$  ist implizit durch die Wahl der Parameter erfüllt (siehe Tabelle 1.2).

Die *Instantiierung* und *Initialisierung* (Facette F2) von Modell 1 findet sich bereits in der SBML-Implementierung (Anhang C, Zeilennummern wie dort) durch die XML-Attribute value und initialAmount, z.B. für den Parameter  $k_4$

```

667 <parameter id="k4" metaid="_840227" value="180"/>

```

und die Entität CP

```

170 <species id="CP" initialAmount="0.75" name="cdc2k-P" metaid="_000005"
    ↪ compartment="cell">

```

Die entsprechende Festlegung des Parameters  $k_4$  könnte in SED-ML über ein entsprechendes changeAttribute vorgenommen werden (dies stammt nicht aus Anhang E):

```

<listOfChanges>
  <changeAttribute
    target="/sbml:sbml:model/sbml:listOfParameters/sbml:parameter[@id='k4'
    ↪ ']/@value"
    newValue="0.75"/>
</listOfChanges>

```



Das Ziel der Änderung wird durch einen Ausdruck in [XPath](#) (*XML Path Language*, [BERGLUND ET AL. 2010](#)) bestimmt. `listOfChanges` ist ein Kindelement von `model`, durch welches das Modell vollständig instantiiert werden kann.

Die *Postprozessierung* ([Facette F3](#)) der Simulationsergebnisse wird in [SED-ML](#) durch [XML](#)-Elemente vom Typ `dataGenerator` festgelegt, z.B. für  $[M]/[CT]$  (Zeilennummern entsprechend [Anhang E](#)):

```

26 <dataGenerator id="M_CT" name="M_CT">
27   <listOfVariables>
28     <variable id="M" name="M" taskReference="task1"
29       target="/sbml:sbml/sbml:model/sbml:listOfSpecies/sbml:species[
    ↪ @id='M']" />
30     <variable id="CT" name="CT" taskReference="task1"
31       target="/sbml:sbml/sbml:model/sbml:listOfSpecies/sbml:species[
    ↪ @id='CT']" />
32   </listOfVariables>
33   <math xmlns="http://www.w3.org/1998/Math/MathML">
34     <apply>
35       <divide />
36       <ci> M </ci>
37       <ci> CT </ci>
38     </apply>
39   </math>
40 </dataGenerator>

```

Dabei werden Ausgaben von `task1` verarbeitet, die eigentliche Berechnung wird in [MathML](#) (*Mathematical Markup Language*, [AUSBROOKS ET AL. 2003](#)) spezifiziert. In `task1` wird die Verbindung des durchzuführenden Simulationsexperiments mit der Modellinstanz hergestellt (Zeilennummern entsprechend [Anhang E](#)):

```

15 <task id="task1" modelReference="modell1" simulationReference="sim1" />

```

Die Ausgabe der finalen Ergebnisse erfolgt dann über `listOfOutputs`, z.B. für die Darstellung von  $[M]/[CT]$  und  $[YT]/[CT]$  gegen  $t$  (Zeilennummern entsprechend [Anhang E](#)):

```

57 <listOfOutputs>
58   <plot2D id="plot1">
59     <listOfCurves>
60       <curve id="curve_0" logX="false" logY="false" xDataReference="time"
    ↪ yDataReference="M_CT" />
61       <curve id="curve_1" logX="false" logY="false" xDataReference="time"
    ↪ yDataReference="YT_CT" />
62     </listOfCurves>
63   </plot2D>
64 </listOfOutputs>

```

Dabei werden als Referenzen die `dataGenerator`-Ausgaben verwendet.

Über die Formalisierung der extrinsischen Funktion von `Modell 1` kann nur wenig gesagt werden. Zum einen fehlen in [TYSON 1991](#) Angaben zu biologischen Experimenten, auf die sich die Verwendung des Modells bezieht (vgl. [Beispiel 2.5](#)). Zum anderen gibt es nur wenige formale Ansätze, die sich für die extrinsische Funktion von Bio-Modellen benutzen lassen. Die Formalisierung der *biologische Fragen* ([Facette F1](#)), die [TYSON](#) beantworten will, wird in [Abschnitt 4.3.3](#) noch einmal thematisiert.



Lediglich die experimentellen Bedingungen lassen sich qualitativ angeben: Die beschriebene Simulation möchte das Verhalten frühembryonaler Zellen abbilden. Diese Situation lässt sich als 'early embryonic cell' ([CL:0000007](#)) charakterisieren. ■

### 4.2.3. Formalisierung des Modell-Verhaltens

#### Intrinsisches Verhalten

Die Ergebnisse von Simulationen liegen als Ausgaben entsprechender numerischer Programme bereits in codierter Form vor. Das betrifft sowohl die *Rohdaten* als auch die *Simulationsresultate* ([Facette B1](#)). Allerdings sind damit zunächst lediglich die eigentlichen Zahlenwerte formal dargestellt. Für die Interpretation solcher Zahlenwerte ist es zusätzlich notwendig, ihre Herkunft anzugeben und sie zusätzlich mit geeigneten physikalischen Einheiten zu versehen. Ohne weitere Angaben stellt z.B. der zweite Wert der zweiten Spalte von [Tabelle 2.2](#) nur die Zahl  $4.292610 \cdot 10^{-4}$  dar. Erst durch Angabe des Entstehungskontextes, d.h. einer vollständigen Beschreibung der Simulation, die diesen Zahlenwert geliefert hat, kann der Wert als Zustandsgröße  $[M]/[CT]$  (mit Einheit 1) des entsprechend instantiierten dynamischen Systems zum Zeitpunkt  $t = 0.1$  s interpretiert werden. Dabei ist dieser Zeitpunkt seinerseits eine (simulationsabhängige) Interpretation des zugehörigen Werts der ersten Spalte von [Tabelle 2.2](#). Die Angabe von Simulationsverhalten muss also immer Bezug nehmen sowohl auf die strukturelle als auch auf die funktionale Bedeutung des Modells.

Es gibt zur Zeit noch keine zufriedenstellende Möglichkeit, Rohdaten und Simulationsresultate mit den entsprechenden strukturellen und funktionalen Informationen zu verbinden. Viele Simulatoren geben die Simulationsergebnisse als Matrizen bzw. Tabellen aus, z.B. codiert als [CSV](#) (*Comma-Separated Values*, [SHAFRANOVICH 2005](#)). Ein Vorschlag, den Spalten und Zeilen von [CSV](#)-Dateien eine Bedeutung zuzuweisen, stellt das [XML](#)-basierte Dateiformat [Fielded Text](#) (<http://www.fieldedtext.org/>) dar. Allerdings kann hier die Bedeutung lediglich durch sprechende Bezeichner ([Name](#)) für die Spalten vergeben werden. Dagegen sieht das ebenfalls [XML](#)-basierte [SBRML](#) (*Systems Biology Results Markup Language*, [DADA ET AL. 2010](#)) die Einbindung von Ontologien zum *Grounding* der Bedeutung der Resultate vor. [SBRML](#) ist auf die Darstellung von Simulationsergebnissen von Bio-Modellen spezialisiert. Die vorliegende Version von [SBRML](#) (Level 1, Version 1) vermischt die Beschreibung von Resultaten mit der Beschreibung von Informationen zur Modellverwendung. Aus diesem Grund wird zur Zeit mit [NuML](#) (*Numerical Markup Language*, <http://code.google.com/p/numl/>) ein neues Dateiformat für numerische Simulationsresultate entwickelt. Ausgangspunkt von [NuML](#) sind entsprechende Fragmente von [SBRML](#). Ziel dieser Entwicklung ist es unter anderem, für Simulationsresultate einen standardisierten Weg für die Referenzierung zugehöriger funktionaler Information (z.B. codiert in [SED-ML](#)) bereitzustellen. Zeitgleich mit der aktuellen Entwicklung von [NuML](#) wird auch an der Unterstützung dieses Dateiformats durch Simulationssoftware gearbeitet, unter anderem in [COPASI](#).

Die *Ausgaben* und die zugehörigen *Charakteristiken* ([Facette B2](#)) von Simulationsresultaten sind wie die Rohdaten zunächst numerisch gegeben. Für die Formalisierung dieser numerischen Ausgaben gilt das oben Gesagte. Für eine geeignete grafische Repräsentation und Visualisierung der Ausgaben gibt es zahlreiche spezialisierte Programme und Beschrei-

bungssprachen. Visualisierung von Resultaten ist jedoch ein eigenständiges Themengebiet, weit außerhalb der hier beschriebenen formalen Repräsentation von Wissen, welches hier nicht weiter verfolgt werden soll.

Als das hier beschriebene Promotionsvorhaben begonnen wurde, gab es in der Systembiologie keine formalen Ansätze für die qualitative Beschreibung des Modellverhaltens (*Dynamik*) und dessen *Diversifikation* (Facette B3). Mit temporalen Logiken (siehe Abschnitt 5.3) können mögliche Modellverhalten charakterisiert werden. Dies wird z.B. in BIOCHAM (*Biochemical Abstract Machine*, CHABRIER-RIVIER ET AL. 2005) ausgenutzt, um temporale Eigenschaften biochemischer Systeme zu untersuchen (vgl. Abschnitt 5.3). Temporale Logiken erlauben jedoch nur die Formulierung sehr einfacher zeitlicher Eigenschaften dynamischer Systeme (für Details siehe Abschnitt 5.3). Weiterhin kann mit der qualitativen Modellierung (vgl. Abschnitt 5.2) Systemverhalten qualitativ beschrieben werden. Allerdings bietet die qualitative Modellierung nur allgemeine Formalismen zur Beschreibung von Verhalten, ohne eine Ontologie von Verhaltensweisen und Verhaltenscharakteristika bereitzustellen. Aus diesen Gründen wurde in Zusammenarbeit mit der BioModels.net-Community mit der Entwicklung einer entsprechenden Ontologie begonnen: TEDDY (*TErminology for the Description of DYnamics*, COURTOT ET AL. 2011). Die in Kapitel 6 ausführlich beschriebene Ontologie TEDDY stellt Konzepte für Verhaltensweisen zur Verfügung, mit deren Hilfe die Dynamik von Modellen qualitativ gefasst werden kann. Außerdem beinhaltet TEDDY Konzepte, mit denen sich die Diversifikation von Dynamik kennzeichnen lässt. Allerdings liefert TEDDY nur das Vokabular für eine propositionale Beschreibung des Modellverhaltens. In Abschnitt 6.3 wird mit DYML (*Dynamics Markup Language*, KNÜPFER ET AL. 2013b) eine neue Sprache skizziert, die insbesondere die Beziehungen zwischen Simulationsbedingungen, also funktionaler Information, und dem resultierenden Modellverhalten erfassen kann. Dabei benutzt DYML Konzepte aus TEDDY.

### Extrinsisches Verhalten

Die quantitativen Ergebnisse von biologischen Experimenten (extrinsische Facette B1 und Facette B2) werden analog zu den oben beschriebenen numerischen Simulationsergebnissen formal dargestellt. Dabei treten auch dieselben Probleme der Interpretation von Zahlenwerten bezüglich der sie erzeugenden Experimente und deren Bedingungen auf, wie sie oben beschrieben sind.

Die qualitativen *biologischen Phänomene* und deren *Variabilität* (Facette B3), die in biologischen Experimenten beobachtet wurden, können durch Annotationen auf entsprechende Datenbanken und Ontologien bestimmt werden. So enthält z.B. der GO-Zweig 'Biological Process' (GO:0008150) entsprechende Konzepte zur Charakterisierung zellulärer Phänomene. Zahlreiche Ressourcen beschreiben Phänotypen und Krankheiten spezieller Organismen. Menschliche Phänotypen sind z.B. in der HPO (*Human Phenotype Ontology*, ROBINSON ET AL. 2010) dargestellt und menschliche Krankheiten in der ICD (*International Classification of Diseases*, <http://www.who.int/classifications/icd/>). Diese Ressourcen können ebenfalls der Beschreibung entsprechender biologischer Phänomene dienen. Für die Spezifikation der Variabilität können Informationen über Phänomene mit den jeweiligen Bedingungen ihres Auftretens verknüpft werden, also mit funktionalen Informationen.

### Beispiel 4.5 (Formalisierung des Verhaltens von Modell 1 – Zeitreihe)

Die in Beispiel 2.7 gezeigten Simulationsergebnisse können wie folgt formalisiert werden. Die

Darstellung der Rohdaten und der finalen Simulationsresultate kann z.B. als [CSV](#) erfolgen, wobei die Spalten und Zeilen durch [Fielded Text](#) spezifiziert werden können. Außerdem können die numerischen Werte und Informationen zu ihrer Interpretation in [SBRML](#) codiert werden. Wegen den oben angeführten Problemen der beiden Dateiformate soll hier auf Details verzichtet werden. Die Formalisierung der *Ausgabe* und der *Charakteristiken* erfolgt auf analoge Weise.

Die *Dynamik* ([Facette B3](#)) der Zeitreihe lässt sich mit [TEDDY](#) wie folgt charakterisieren: Das dynamische System befindet sich in einem 'stable limit cycle' ([TEDDY\\_0000114](#)), was sich als harmonische Oszillation ('single-periodic oscillation', [TEDDY\\_0000065](#)) beobachten lässt.

Die Formalisierung der (extrinsischen) quantitativen Ergebnisse der biologischen Experimente erfolgt wie bei den numerischen Simulationsergebnissen. Eine Diskussion der Formalisierung der *biologischen Phänomene* der TYSON-Modelle erfolgt im nächsten [Beispiel 4.6](#). ■

#### Beispiel 4.6 (Formalisierung der Dynamik von Modell 2)

In diesem Beispiel soll es insbesondere um die Formalisierung der Dynamik von Modell 2 gehen und um die Verbindung von intrinsischer Dynamik mit extrinsischen Phänomenen.

[TYSON 1991](#) identifiziert drei unterschiedliche *Dynamiken* ([Facette B3](#)) abhängig von den gewählten Parametern  $k_4$  und  $k_6$  (vgl. [Abbildung 1.4](#)). Diese drei Modi A, B und C von Modell 2 (vgl. [Abschnitt 1.2.3](#)) können analog zum [Beispiel 4.5](#) mit [TEDDY](#) charakterisiert werden:

Modus A ist ein stabiler Fixpunkt ('stable spiral' [[TEDDY\\_0000126](#)]),

Modus B ist eine Grenzzyklus-Oszillation ('stable limit cycle' [[TEDDY\\_0000114](#)]) und

Modus C ist ein erregbarer Schalter ('excitability' [[TEDDY\\_0000179](#)]).

Die *Diversifikation* ([Facette B3](#)), die durch den Übergang zwischen dem stabilen Fixpunkt-Verhalten und dem oszillierendem Grenzzyklus-Verhalten gekennzeichnet ist, kann ebenfalls mit [TEDDY](#) charakterisiert werden als 'supercritical Hopf bifurcation' ([TEDDY\\_0000074](#)). Dieser Übergang zwischen zwei Dynamiken und die Verbindung zu den entsprechenden Parameterbereichen kann mit [TEDDY](#) allein nicht beschrieben werden. In [Abschnitt 6.3](#) wird gezeigt, wie eine Formalisierung mit einer noch zu entwickelnden Sprache (Arbeitstitel [DYML](#)) aussehen könnte (vgl. auch [Beispiel 6.3](#)).

Die extrinsische Bedeutung der Dynamiken von Modell 2 wird durch entsprechende Annotationen festgelegt. Wie in [Abschnitt 1.2.3](#) beschrieben, ordnet [TYSON 1991](#) dem stabilen Fixpunkt (Dynamik) die Hemmung der Metaphase in unbefruchteten Eizellen (biologisches Phänomen) zu. Das Phänomen kann durch eine entsprechende Annotation auf 'cell cycle arrest' ([GO:0007050](#)) bestimmt werden. Durch Verbindung der Phänomene mit den entsprechenden Bedingungen können auch Aussagen zur *Variabilität* erfasst werden. So kann etwa das Phänomen 'cell cycle arrest' ([GO:0007050](#)) in einer 'early embryonic cell' [CL:0000007](#) beobachtet werden. Das entsprechende [RDF-Triple](#)<sup>3</sup> sieht folgendermaßen aus:

---

<sup>3</sup>In dieser Arbeit werden alle [RDF](#)-Ausdrücke in [Turtle](#) (*Terse RDF Triple Language*, [BECKETT ET AL. 2013](#)) angegeben.

```
@prefix dc: <http://purl.obolibrary.org/obo/> .

<http://identifiers.org/go/G0:0007050>
  dc:BF0_0000066 <http://identifiers.org/cl/CL:0000007> .
```

Die dabei verwendete Relation 'occursIn' (BF0\_0000066) aus der RO (*OBO Relations Ontology*, <http://code.google.com/p/obo-relations/>) spezifiziert eine Beziehung zwischen einem Prozess und einem Objekt, in welchem der Prozess stattfindet. Im Beispiel ist das Objekt eine Zelle im frühen Embryonalstadium, d.h. das modellierte System „Zellzyklus“ (strukturelle Information) unter bestimmten Randbedingungen und in einem bestimmten Zustand (funktionale Information). ■

### 4.3. Computergestützte Bio-Modellierung im wissenschaftlichen Kontext

Die Konstruktion und Verwendung von Bio-Modellen erfolgt im wissenschaftlichen Kontext biologischer Fragestellungen (vgl. Abschnitt 1.3.1). Für die Etablierung computergestützter Modellierung und Simulation muss die formale Repräsentation relevanter Wissensfragmente diesen Kontext berücksichtigen. In diesem Abschnitt sollen Aspekte des wissenschaftlichen Kontextes diskutiert werden, die eine entscheidende Rolle bei der Realisierung der Computerunterstützung spielen. Neben den globalen Metainformationen (Abschnitt 4.3.1) sind die Intentionen der Modellverwendung (Abschnitt 4.3.2) in geeigneter Weise formal zu repräsentieren. Um diese teleologische Funktion von Bio-Modellen formal repräsentieren zu können, wird abschließend mit MINTENTO (*Modelling Intention Ontology*, KNÜPFER ET AL. 2013a) eine entsprechende Ontologie skizziert (Abschnitt 4.3.3).

#### 4.3.1. Formalisierung der globalen Metainformationen von Bio-Modellen

Die globalen Metainformationen eines Bio-Modells (Abschnitt 2.1.2) beschreiben die Herkunft des Modells, den Zugriff auf das Modell und die Beziehungen zu anderen Modellen. Für eine Computerunterstützung ist auch dieses Wissen geeignet zu formalisieren.

Die Definition der *Herkunft* (Facette M1) eines Modells umfasst eine Beschreibung seiner Autoren und des Datums seiner Erstellung. Es gibt verschiedene Dienste, die versuchen, Autoren eindeutig zu identifizieren, z.B. ORCID (*Open Researcher and Contributor ID*, BUTLER 2012). Das Datum der Erstellung sollte in einem standardisierten Datumsformat angegeben werden, z.B. gemäß der ISO 8601 (ISO 2004).

Für die genaue Bestimmung der Herkunft wird ein Bio-Modell einer *Publikation* (Facette M1) zugeordnet, falls dies möglich ist. Dies kann zum einen durch eine Literaturangabe erfolgen, meist als Verweis auf einen Eintrag in einer Literaturdatenbank. Für Literaturangaben existieren verschiedene Standardformate, wie etwa RIS (*Research Information System Format*, [http://www.refman.com/support/risformat\\_intro.asp](http://www.refman.com/support/risformat_intro.asp)) oder das Dateiformat von BibTeX (PATASHNIK 1988). Zum anderen wird heutzutage zunehmend direkt auf das Dokument in Online-Journalen verwiesen, etwa über eine DOI (*Digital Object Identifier*, PASKIN 2010) oder über eine URL. Es gibt auch Online-Kataloge für Primärliteratur, wie etwa PubMed (<http://www.ncbi.nlm.nih.gov/pubmed/>) für medizinische und biologische Literatur. Die Referenzierung erfolgt über eine PMID (*PubMed Identifier*).

Als Qualifikatoren für Herkunftsinformationen über ein Bio-Modell bietet sich das Vokabular **Dublin Core** (KUNZE ET AL. 2007) an, z.B. **dc.creator** für die Autoren. Diese Qualifikatoren können als Prädikate in entsprechenden **RDF**-Repräsentationen der Metainformationen eingesetzt werden.

Die Formalisierung von Angaben über die *Implementierung* (Facette M2) eines Bio-Modells und über den *Zugriff* (Facette M2) auf das implementierte Modell kann teilweise ähnlich erfolgen wie bei den Angaben zur Herkunft. Dies betrifft Angaben über beteiligte Personen bei Implementierung und Kuration und die zugehörigen Zeitangaben. Die Angabe des Dateiformats erfolgt bei **XML**-basierten Sprachen entweder als Datentypdeklaration, durch Verweis auf ein **XML**-Schema oder durch eine Namensraumangabe für das Wurzelement. Die rechtlichen Bedingungen können z.B. mit **dc.rights** und einem Verweis auf die entsprechenden Rechteinhaber und Lizenzbedingungen spezifiziert werden.

Die *Beziehungen* (Facette M3) eines Bio-Modells zu anderen Modellen kann mit Hilfe von Qualifikatoren ausgedrückt werden, z.B. die Teil-Ganzes-Beziehung mit **hasPart**, eine Ähnlichkeit mit **isVersionOf** und die Weiterentwicklung mit **isDerivedFrom** (vgl. **BioModels.net Qualifiers**). Die Details einer Vereinfachung oder Integration sind mit diesen Mitteln jedoch nicht zu fassen. Für Vereinfachungen müssten die zugrundeliegenden Annahmen und die Vereinfachungsschritte formal beschrieben werden. Für die Formalisierung von Annahmen kann **CLP** nützlich sein. Bei der Integration von Bio-Modellen müssen insbesondere die Schnittstellen zwischen den Modellen und Synchronisationsbedingungen festgelegt werden. Die Formalisierung dieser Informationen und die formale Beschreibung von Vereinfachungsschritten stellen offene Problem dar.

Es gibt erste Ansätze der Versionskontrolle für Bio-Modelle (z.B. **WALTEMATH ET AL. 2013** und **MILLER ET AL. 2011**) zur Beschreibung der *Versionsgeschichte* (Facette M3). Für die Differenz zwischen verschiedenen Versionen eines Bio-Modells vergleichen dabei **WALTEMATH ET AL. 2013** zunächst die Netzwerke der Modellversionen. Die Differenz der Netzwerke wird durch Operationen (Hinzufügung und Löschung von Knoten und Kanten) beschrieben, die ein Netzwerk in das andere überführen. Zusätzlich werden Änderungen im dynamischen System betrachtet, etwa ein Update eines kinetischen Gesetzes.

#### Beispiel 4.7 (Formalisierung der globalen Metainformationen vom Modell 1 und Modell 2)

Die Modelle werden in **TYSON 1991** beschrieben. Diese *Publikation* (Facette M1) findet sich in PubMed (<http://identifiers.org/pubmed/1831270>). Für den Autor John J. Tyson (*Herkunft*, Facette M1) ist keine standardisierte ID bekannt.

Der *Zugriff* (Facette M2) auf die in **SBML** implementierten Modelle erfolgt wie in **Abschnitt 2.1.2** beschrieben über die **BioModels Database**. Das Dateiformat ist über die Namensraumangabe des Wurzelements (**sbml**) gegeben:

```
xmlns="http://www.sbml.org/sbml/level2/version4"
```

Es handelt sich also um eine *Implementierung* (Facette M2) in **SBML** Level 2, Version 4.

Das Modell ist unter der Lizenz „CC0 Public Domain Dedication“ in der **BioModels Database** verfügbar. Die Lizenz von Modell 1 kann durch folgenden **RDF**-Ausdruck beschrieben werden:

```
@prefix biomodels: <http://identifiers.org/biomodels.db/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix cc: <http://creativecommons.org/publicdomain/> .

BIOMD0000000005 dc:license cc:zero\1.0 .
```

Für die formale Angabe der in [Abschnitt 2.1.2](#) beschriebenen *Beziehung* ([Facette M3](#)) von Modell 2 zum „Brüsselator“ fehlt eine Referenz auf eine Implementierung des Modells von [NICOLIS ET AL. 1977](#). Die Ableitung von Modell 2 aus Modell 1 kann in [RDF](#) so ausgedrückt werden:

```
@prefix biomodels: <http://identifiers.org/biomodels.db/> .
@prefix bqmodel: <http://biomodels.net/model-qualifiers/> .

biomodels:BIOMD0000000005
  bqmodel:isDerivedFrom biomodels:BIOMD0000000006 .
```

Für eine genauere formale Beschreibung der Ableitung von Modell 2 fehlen die sprachlichen Mittel.

Eine *Versionsgeschichte* ([Facette M3](#)) ist für die beiden TYSON-Modelle nicht verfügbar. ■

#### 4.3.2. Teleologische Funktion als Treiber für die Computerunterstützung

Die Funktion eines Bio-Modells beschreibt, warum und wie es verwendet werden soll. Der zentrale Treiber der Modellverwendung, und damit ihr Ausgangspunkt, sind dabei die biologischen Fragen ([Facette F1](#)), die mit Hilfe des Modells beantwortet werden sollen. Dies kann man auch als Intention der Modellverwendung bezeichnen. Die Intention bestimmt darüber, was sinnvolle Verwendungen sind. Extrinsische teleologische Funktion (Intentionen) determiniert also intrinsische teleologische Funktion (intendierte Verwendung). Somit sind für die Computerunterstützung des Modellentwicklungszyklus die Intentionen und die dadurch bestimmten intendierten Verwendungen die zentralen Steuerungselemente. Eine geeignete formale Repräsentation teleologischer Funktion ist somit eine wichtige Voraussetzung für die computergestützte Modellierung und Simulation.

Wie kann man sich konkret eine solche Computerunterstützung auf der Basis teleologischer Funktion vorstellen? Die folgenden Aufgaben sollen von dem Potential einer formalen Repräsentation der Intentionen und intendierten Verwendungen von Bio-Modellen einen Eindruck vermitteln. Dabei wird im Folgenden von „Verwendungsmethoden“, kurz „Methoden“ die Rede sein. Damit sind Arten der intendierten Verwendung von Bio-Modellen gemeint, also z.B. die Simulation einer Zeitreihe.

**Methodenauswahl** Geeignete Verwendungsmethoden werden nach den Intentionen, der Art der verfügbaren Daten und den gewünschten Resultaten ausgewählt.

**Vorschlag alternativer Methoden** Alternative Verwendungsmethoden können nach Ähnlichkeit vorgeschlagen werden.

**Spezifikation der Verwendung** Die Spezifikation von Simulationsexperimenten (z.B. in [SED-ML](#)) wird unterstützt. Dabei wird die Konsistenz bezüglich Intentionen und vorhandenen Daten gesichert.



**Methoden-Komposition** Anhand der notwendigen Eingabedaten und der möglichen Resultate von Methoden können ganze Methoden-Workflows erstellt werden, die der Erfüllung der Intentionen dienen.

**Prüfung der Anwendbarkeit** Methoden haben oft bestimmte strukturelle Anforderungen an die Modelle, auf die sie angewendet werden. Sind diese Anforderungen durch ein Modell nicht erfüllt, können geeignete Modellmodifikationen vorgeschlagen werden.

Um diese Aufgaben zu lösen, ist eine Sammlung von Verwendungsmethoden notwendig. Eine solche Methodensammlung könnte in der angestrebten Ontologie [MINTENTO](#) enthalten sein (siehe [Abschnitt 4.3.3](#)). Für jede Methode müssen dabei folgende Informationen spezifiziert werden:

- Intentionen der Modellverwendung, die durch diese Methode realisiert werden,
- notwendige Daten für die Benutzung der Methode,
- Art der Resultate, die die Methode liefert und
- strukturelle Eigenschaften des Modells, die für die Benutzung der Methode vorliegen müssen.

Die so beschriebenen Methoden finden sich in den unteren Ebenen des [MINTENTO](#)-Zweiges 'intended use'; die oberen Ebenen dieses Zweiges dienen ihrer Klassifikation (siehe [Abschnitt 4.3.3](#)). Verwendungsmethoden in dem hier benutzten Sinne sind nicht zu verwechseln mit Simulationsalgorithmen, wie sie z.B. von [KiSAO](#) (*Kinetic Simulation Algorithm Ontology*, [COURTOT ET AL. 2011](#)) spezifiziert werden. Eine Verwendungsmethode, wie z.B. 'time-course simulation' beschreibt, *was* mit einem Modell gemacht werden soll, während ein Algorithmus festlegt, *wie* etwas gemacht werden soll.

Eine weitere Anwendung einer durch [MINTENTO](#) bereitgestellten Methodensammlung liegt in der ontologischen Fundierung der Simulationstypen in [SED-ML](#). Jeder Simulationstyp in [SED-ML](#) kann einer Methode in [MINTENTO](#) zugeordnet werden. Die so verfügbar gemachten Informationen über die Methoden können zur Spezifikation von Simulationsexperimenten und deren Konsistenzprüfung verwendet werden. Außerdem kann die Konvertierung in andere formale Sprachen für die Beschreibung von Simulationsexperimenten unterstützt werden.

#### 4.3.3. Entwurf einer *Modelling Intention Ontology* (MINTENTO)

Hier sollen erste Ideen einer Ontologie für die teleologische Funktion von Bio-Modellen vorgestellt werden, die als Grundlage für die Umsetzung der im vorherigen [Abschnitt 4.3.2](#) skizzierten Computerunterstützung bei der Modellverwendung dienen kann. Die Darstellung der *Modelling Intention Ontology* ([MINTENTO](#)) folgt dabei [KNÜPFER ET AL. 2013a](#). [MINTENTO](#) beinhaltet sowohl extrinsische Intentionen der Modellverwendung als auch intrinsische intendierte Verwendungen, die zur Realisierung der Intentionen beitragen (können). Diese Zweiteilung in „function“ (hier als „Intention der Modellverwendung“ bezeichnet) und deren „realisation“ (hier: „intendierte Verwendung“) wird auch in [BUREK ET AL. 2006](#) verwendet.



Die entsprechenden Zweige von **MINTENTO** sind 'model use intention' (Intention der Modellverwendung) und 'intended use' (intendierte Verwendung). Dabei ist im Sinne von **BFO** (*Basic Formal Ontology*, **GRENON ET AL. 2004**) die 'model use intention' eine 'function' (**BFO\_0000034**) und 'intended use' ein 'process' (**BFO\_0000015**). Die Verbindung von Intentionen der Modellverwendung und intendierten Verwendungen, die zu deren Realisierung beitragen können, erfolgt über die Relation **realisedBy** (siehe unten).

#### Intentionen der Modellverwendung

**MORRISON ET AL. 1999** beschreiben ausführlich verschiedene übergeordnete Funktionen, die ein Modell erfüllen kann. Zusammenfassend werden die Arten teleologischer Funktion (**Facette F1**) eines Modells wie folgt charakterisiert:

„[...] models fulfil a wide range of functions in building, exploring and applying theories; in various measurement activities; and in the design and production of technologies for intervention in the world.“ (**MORRISON ET AL. 1999**, S. 24)

Diese Aufzählung kann als Startpunkt für eine Klassifikation genereller Intentionen der Modellverwendung dienen. Nach **MORRISON ET AL. 1999** wird ein Modell benutzt, um (**MINTENTO**-Konzepte in Klammern):

- eine Theorie zu konstruieren ('to construct'),
- eine Theorie zu untersuchen ('to explore'),
- etwas zu messen ('to measure') und
- in etwas einzugreifen ('to intervene').

Die Intentionen der Modellverwendung sind dabei basierend auf zwei Unterscheidungen klassifiziert: Zum einen vermittelt ein Modell zwischen Theorie und Wirklichkeit („mediate between theory and the world“, **MORRISON ET AL. 1999**, S. 11), es wird also entweder verwendet, um eine Theorie zu beeinflussen ('to affect theory') oder auf die Realität einzuwirken ('to affect reality'). Zum anderen gibt es eine Unterscheidung, ob ein Modell etwas beeinflussen soll ('to effect') oder ob es dazu dienen soll, etwas zu lernen ('to learn') (**MORRISON ET AL. 1999**, S. 11). Die oberen Ebenen des **MINTENTO**-Zweiges 'model use intention' können mit Hilfe dieser beiden Unterscheidungen wie in **Abbildung 4.3** dargestellt werden.

Für eine detaillierte Klassifikation von Intentionen der Modellverwendung müsste zunächst eine Datenerhebung erfolgen, etwa durch Analyse von Publikationen, in welchen Bio-Modelle zur Beantwortung biologischer Fragen eingesetzt werden. Dies ist im Rahmen der vorliegenden Arbeit nicht möglich. Erste Überlegungen zeigen aber, dass Bio-Modelle im akademischen Umfeld oft dazu eingesetzt werden, biologische Systeme zu verstehen. Das Ziel ist also die Erweiterung der biologischen Theorie. Dafür muss die bestehende Theorie untersucht ('to explore') und geeignet modifiziert ('to construct') werden. Im pharmakologischen Umfeld werden Bio-Modelle dagegen auch eingesetzt, um Auswirkungen bestimmter Veränderungen auf ein biologisches System vorherzusagen, also in das reale System einzugreifen ('to intervene'). Dagegen findet die Verwendung eines Modells zur Messung ('to measure') in der Bio-Modellierung keine unmittelbare Anwendung.

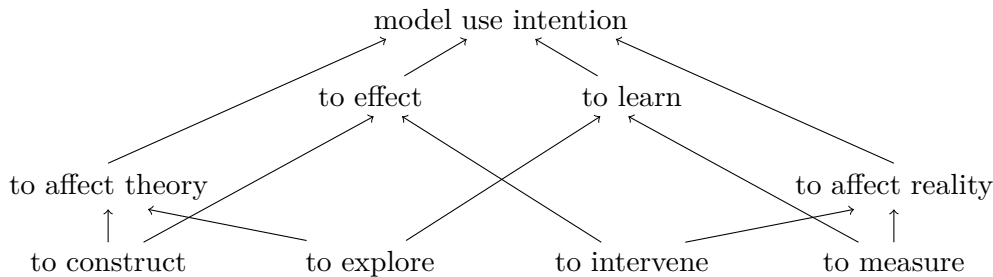


Abbildung 4.3.: Obere Ebenen des [MINTENTO](#)-Zweiges 'model use intention'. Die Pfeile repräsentieren Subsumtionsbeziehungen (*is-a*).

Beschränken wir uns hier auf die Verwendung von Bio-Modellen zur Erklärung biologischer Beobachtungen, also auf die Erweiterung der Theorie, können typische Aufgaben von Bio-Modellen angegeben werden:

- ein beobachtetes Verhalten zu approximieren ('to approximate'),
- die prinzipielle Möglichkeit eines bestimmten Verhaltens zu zeigen ('to demonstrate'),
- die Arten möglicher Dynamiken zu untersuchen ('to investigate') und
- den Einfluss von Parametern auf das Verhalten zu untersuchen ('to analyse').

Dabei wird bei den ersten beiden Aufgaben das Modell validiert. Das ist ein wichtiger Schritt bei der Theoriekonstruktion. Die beiden letzten Aufgaben explorieren eine bestehende Theorie. Die Trennlinie zwischen beiden Intensionsarten ist dabei nicht scharf: Die Entwicklung einer Theorie benötigt meist ein iteratives Vorgehen aus Validierung und Exploration. An dieser Stelle ist eine weitere, intensivere Auseinandersetzung mit den Intentionen für die Verwendung von Bio-Modellen notwendig.

Die Intentionen von [TYSON 1991](#) (S. 7329) für die Verwendung der beiden Modelle (vgl. [Abschnitt 1.2.3](#)) dienen sicherlich einer Erweiterung der Theorie, genauer einem Verstehen der Kontrolle des Zellzyklus. Dabei soll die prinzipielle Fähigkeit der Modelle gezeigt werden ('to demonstrate'), bestimmte Phänomene zu erklären. In der Phasenraumanalyse untersucht [TYSON 1991](#) dafür mögliche Dynamiken des Modells ('to analyse') und bezieht diese auf die zu erklärenden Phänomene. Die Exploration dient also dem höheren Ziel der Validierung.

#### Intendierte Verwendung von Bio-Modellen

Die Hauptverwendung von Bio-Modellen liegt in der Simulation ihres dynamischen Verhaltens. Es gibt auch die Möglichkeit, bestimmte dynamische Eigenschaften eines Modells durch mathematische Analyseverfahren zu gewinnen, ohne das zeitliche Verhalten explizit zu betrachten, z.B. die Flux Balance Analysis ([ORTH ET AL. 2010](#)). Der [MINTENTO](#)-Zweig 'intended use' kann also in 'simulation' und 'analysis' unterteilt werden. Simulationen können weiter eingeteilt werden in einfache Simulationen 'elementary simulation' und kombinierte Simulationen 'combined simulation' (vgl. [Abschnitt 1.1.2](#)). Die oberen Ebenen von 'intended use' stellen sich also wie in [Abbildung 4.4](#) dar.

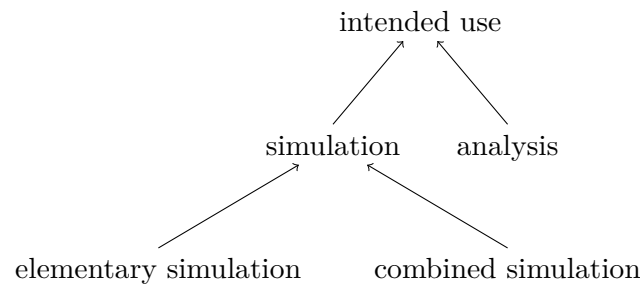


Abbildung 4.4.: Obere Ebenen des **MINTENTO**-Zweiges 'intended use'. Die Pfeile repräsentieren Subsumtionsbeziehungen (**is-a**).

Ein Beispiel für eine 'elementary simulation' ist die Simulation einer Zeitreihe ('time-course simulation'). Die Bestimmung der Dynamik unter Variation von Parametern ('parameter scan') ist ein Beispiel für eine kombinierte Simulation. Damit können z.B. Bifurkationsdiagramme erstellt werden. Eine weitere Detaillierung des **MINTENTO**-Zweiges 'intended use' soll hier nicht vorgenommen werden.

#### Realisierung von Intentionen der Modellverwendung durch intendierte Verwendungen

Die Relation **realisedBy** besteht zwischen einem bestimmten Ziel ('model use intention') und einer Verwendungsart eines Bio-Modells ('intended use'), die der Erreichung dieses Ziels dienen kann: Für ein Unterkonzept *A* von 'model use intention' und ein Unterkonzept *B* von 'intended use' gilt „*A realisedBy B*“, falls es für jede Instanz *a* von *A* eine Instanz *b* von *B* gibt, so dass die Ausführung des Prozesses *b* das Ziel *a* realisiert (**realises**). Die entsprechende Instanzlevel-Relation **realises** kann dabei auch abgeschwächt definiert werden als „trägt bei“.

Für die Illustration solcher **realisedBy**-Relationen sollen zwei Beispiele angeführt werden (für die entsprechenden Intentionen der Modellverwendung siehe oben):

- 'to approximate' realisedBy 'time-course simulation',
- 'to analyse' realisedBy 'parameter scan'.

Eine genaue Bestimmung dieser Relation kann erst erfolgen, wenn die entsprechenden **MINTENTO**-Zweige ausgebaut sind. Hier ging es zunächst um eine Vermittlung der grundsätzlichen Idee.

## Zusammenfassung

Die formale Repräsentation eines Bio-Modells bezüglich der Wissensfacetten ist ein Fundament für die computergestützte Modellierung und Simulation. Für eine umfassende Computerunterstützung muss eine solche Repräsentation die Struktur, die Funktion und das Verhalten eines Bio-Modells umfassen und eine Beziehung zwischen der formalen Semantik und der biologischen Bedeutung herstellen. In diesem Kapitel wurden Bedingungen für eine computergestützte Bio-Modellierung untersucht ([Abschnitt 4.1](#)) und gezeigt, wie mit bestehenden Mitteln die Wissensfacetten eines Bio-Modells formalisiert werden können ([Abschnitt 4.2](#)). Bei dieser Darstellung wurde auch auf Wissensaspekte eines Bio-Modells hingewiesen, für die es momentan keine adäquate formale Sprache oder Ontologie gibt (vgl. [Tabelle 4.1](#) auf [Seite 147](#)). Dieses formale Fundament kann sein volles Potential erst im wissenschaftlichen Kontext entfalten, in dem ein Bio-Modell verwendet wird. Mit dem wissenschaftlichen Kontext eines Bio-Modells und dessen formaler Rekonstruktion beschäftigt sich der abschließende [Abschnitt 4.3](#).

Bei Betrachtung von [Tabelle 4.1](#) ([Seite 147](#)) fällt zunächst auf, dass insbesondere die Formalisierung der extrinsischen, biologischen Wissensfacetten mit existierenden Mitteln nur unzureichend möglich ist. Das liegt zum einen an den in [Abschnitt 2.1.3](#) angesprochenen Problemen, die biologischem Wissen immanent sind. Zum anderen ist die schwächere Ausprägung von formalen Ansätzen für die biologische Bedeutung von Bio-Modellen wissenschaftshistorisch zu begründen: Die Biologie hat sich erst in den letzten Jahren den mathematischen Methoden der Modellierung und Simulation geöffnet.

Dagegen sind die Entwickler von Computer-Simulatoren mit formalen Darstellungsmitteln vertraut. Zur Entwicklung standardisierter Mittel für die formale Repräsentation intrinsischer Modellbedeutung war eine vergleichbar kleine Schwelle zu überwinden. Mit [SBML](#) und [CellML](#) stehen exzellente formale Sprachen für die Beschreibung der intrinsischen Modellstruktur zur Verfügung. Darüber hinaus ist die aktuelle Entwicklung von [SED-ML](#) als formale Sprache für die Beschreibung von Simulationsexperimenten vielversprechend.

Ein weiteres bedeutendes Defizit liegt in der Formalisierung der teleologischen Funktion. Die teleologische Funktion stellt den zentralen Treiber jeder Modellierung dar. Durch sie werden die biologischen Fragen (Intentionen) über das Zielsystem in Verwendungsweisen des Modells übersetzt. Die Verwendung eines Modells in Simulationsexperimenten wiederum generiert erst dynamisches Modellverhalten aus der Modellstruktur. In [Abschnitt 4.3.2](#) wird beschrieben, wie die Formalisierung der teleologischen Funktion einen wichtigen Beitrag für die Computerunterstützung leisten kann. Als erster Schritt in Richtung einer entsprechenden formalen Beschreibung teleologischer Funktion wird in [Abschnitt 4.3.3](#) eine Ontologie entworfen, mit der die Intentionen und Verwendungsweisen von Bio-Modellen spezifiziert werden können.

Das größte offensichtliche Defizit liegt in der formalen, propositionalen Beschreibung der Dynamik von Bio-Modellen. Um dieses Defizit zu beseitigen, liegt ein Schwerpunkt der vorliegenden Arbeit auf der formalen Repräsentation des dynamischen Verhaltens von Bio-Modellen. Das folgende [Kapitel 5](#) stellt entsprechende Techniken für die qualitative Beschreibung von Modellverhalten vor. Im abschließenden [Kapitel 6](#) wird mit [TEDDY](#) eine Ontologie für die Dynamik von Bio-Modellen entwickelt.

Formales Mittel	Sprachelement/ Ontologiezweig	Struktur			Funktion			Verhalten		
		(S1)	(S2)	(S3)	(F1)	(F2)	(F3)	(B1)	(B2)	(B3)
Intrinsische Bedeutungsfacetten										
SBML	species	x			x			x		
	reaction	x								
	kineticLaw	x								
	initialAmount									
	parameter									
CellML	component	x			x			x		
	variable	x								
	connection	x								
	initial_value									
SBO	“modelling framework”	x			x			x		
	“mathematical expression”	x								
SED-ML	Simulation				x			x		
	Change				x			x		
KiSAO	“modeling and simulation algorithm”				x			x		
DYML	feature									
TEDDY	constraints				x			x		
	“temporal behaviour”									
	“behaviour diversification”							x		
qual. Modellierung								x		
temp. Logiken								x		
SBRML	Result							x x		
Fielded Text								x x		
Extrinsische Bedeutungsfacetten										
BioPAX	“PhysicalEntity”	x			x			x		
	“Interaction”	x								
SBGN	“Entity Pool Nodes”	x			x			x		
	“Process Nodes”	x								
UniProt		x			x			x		
Taxonomy Database		x			x			x		
GO	“Biological Process”	x			x			x		
SBO	“physical entity representation”	x			x			x		
	“systems description parameter”	x								
	“occurring entity representation”	x								
	“mathematical expression”	x								
Reactome		x			x			x		
CL					x			x		
SABIO-RK					x			x		
FuGE	Material				x			x x		
	Investigation				x					
	Data									
SBRML	Result				x			x		
Fielded Text					x			x		

Tabelle 4.1.: In [Kapitel 4](#) angeführte formale Mittel zur Beschreibung der Wissensfacetten. Bedeutungen der Akronyme und Referenzen der formalen Mittel finden sich in [Anhang A](#). Abkürzungen: qual. Modellierung = qualitative Modellierung (siehe [Abschnitt 5.2](#)), temp. Logiken = temporale Logiken (siehe [Abschnitt 5.3](#)).



## Teil III.

# Dynamik von Bio-Modellen

„Rather, we would like to know which classes of qualitatively-different behaviors are compatible with the structure of the system, given certain constraints on the values of the parameters and initial conditions.“

---

([DE JONG 2004](#): *Qualitative simulation and related approaches for the analysis of dynamic systems*, S. 93)

Der dritte und letzte Teil der Arbeit widmet sich der formalen Repräsentation der Dynamik von Modellen. Zunächst werden Möglichkeiten vorgestellt, Verhalten qualitativ zu beschreiben ([Kapitel 5](#)). Dabei wird die Eignung der Theorie dynamischer Systeme, der qualitativen Modellierung und der temporalen Logiken für eine Formalisierung der Dynamik von Bio-Modellen untersucht. Dabei zeigt sich, dass die Theorie dynamischer Systeme prinzipiell adäquate Beschreibungsmittel zur Verfügung stellt. Es existiert jedoch keine formale Sprache, um entsprechende Dynamikbeschreibungen in computerverarbeitbaren Form bereitzustellen. Als Vokabular für eine solche formale Sprache der Dynamik von Bio-Modellen wird mit [TEDDY](#) eine Ontologie der Theorie dynamischer Systeme entwickelt ([Kapitel 6](#)).





## 5. Qualitative Beschreibung von Dynamik

„Qualitative modeling concerns representation and reasoning about continuous aspects of entities and systems in a symbolic, human-like manner.“

---

(FORBUS 2008: *Qualitative Modeling*, S. 361)

Die Resultate der Simulation eines Bio-Modells ([Abschnitt 2.2.3](#)) sind numerischer Natur. Damit solche quantitativen Resultate für die Erklärung biologischer Phänomene von Nutzen sein können, müssen sie geeignet qualitativ charakterisiert werden. In diesem Kapitel werden spezielle Techniken vorgestellt, die sich mit der qualitativen Beschreibung von dynamischen Systemen und deren Verhalten auseinandersetzen.

Die Theorie dynamischer Systeme ([Abschnitt 5.1](#)) beschäftigt sich mit der mathematischen Charakterisierung des Verhaltens dynamischer Systeme. Sie stellt dabei Konzepte für relevante Eigenschaften des Verhaltens dynamischer Systeme und für die Diversifikation von Verhalten bereit. Bei der qualitativen Modellierung und Simulation ([Abschnitt 5.2](#)) handelt es sich um ein klassisches Gebiet der Künstlichen Intelligenz, das sich mit der symbolischen Repräsentation physikalischer Prozesse und deren Simulation beschäftigt. Im Hinblick auf die Beschreibung von Dynamik liegt die Stärke der qualitativen Modellierung vor allem in der propositionalen Repräsentation von Systemzuständen. Auch die temporalen Logiken ([Abschnitt 5.3](#)), welche überwiegend modallogische Erweiterungen der klassischen Logik um zeitliche Aspekte darstellen, repräsentieren Systemzustände qualitativ-propositional. Darüber hinaus gestatten temporale Logiken die Formulierung von zeitlichen Aussagen über Folgen von Zuständen. Mit Hilfe des sogenannten „model checkings“ können solche Aussagen über das Verhalten eines dynamischen Systems verifiziert werden. Die Charakterisierung der Dynamik eines Modells mit temporalen Logiken ist nicht besonders detailreich. Sowohl den temporalen Logiken als auch der qualitativen Modellierung fehlen insbesondere die Mittel, Diversifikationen von Dynamik ([Abschnitt 2.2.3](#)) angemessen zu beschreiben.

### 5.1. Theorie dynamischer Systeme

Die Theorie dynamischer Systeme ist eine „qualitative Theorie“ ([JETSCHKE 1989](#), S. 6) des Verhaltens mathematischer Modelle. Durch die Theorie werden Methoden und Konzepte bereitgestellt, mit denen die wesentlichen Merkmale des Systemverhaltens analysiert und charakterisiert werden können:

„We would like to study general (qualitative) features of the behavior of dynamical systems, in particular, to classify possible types of their behavior and compare the behavior of different dynamical systems.“ ([KUZNETSOV 1998](#), S. 39)

Die wichtigsten Anwendungsgebiete der Theorie dynamischer Systeme sind die Theorie nicht-linearer Dynamik und die Chaostheorie. Häufig werden diese Anwendungsgebiete mit der umfassenderen Theorie dynamischer Systeme gleichgesetzt.

Die folgende Darstellung der Theorie dynamischer Systeme richtet sich nach [JETSCHKE 1989](#). Dabei wird jedoch auf eine präzise mathematische Definition der Begriffe verzichtet. Ähnliche Präsentationen der Grundbegriffe der Theorie dynamischer Systeme finden sich in den Einleitungen einschlägiger Lehrbücher zum Thema, wie etwa [STROGATZ 2001](#), [KUZNETSOV 1998](#) und [HALE ET AL. 1991](#).

Nach Einführung zentraler Begriffe der Theorie dynamischen Systems ([Abschnitt 5.1.1](#)) wird sich mit der Charakterisierung zeitlichen Verhaltens in dieser Theorie beschäftigt ([Abschnitt 5.1.2](#)). Anschließend werden qualitative Änderungen des Verhaltens eines dynamischen Systems, sogenannte Bifurkationen, betrachtet ([Abschnitt 5.1.3](#)). Modell 2 wird zur Illustration der Theorie dynamischer Systeme verwendet ([Beispiel 5.1](#) und [Beispiel 5.2](#)).

### 5.1.1. Dynamische Systeme

Zunächst müssen einige Grundbegriffe eingeführt werden. Ein *dynamisches System* ist ganz allgemein eine Menge von Zuständen (*Phasenraum* oder Zustandsraum genannt) und eine Regel für zeitliche Änderungen des Zustands, die auch als *Phasenfluss* bezeichnet wird. Ein Phasenfluss ist dabei eine Familie von Abbildungen auf dem Phasenraum. Für jeden (Anfangs-)Punkt des Phasenraums beschreibt eine dieser Abbildungen die zeitliche Entwicklung. Dabei wird zwischen zeitkontinuierlichen und zeitdiskreten Systemen unterschieden, wobei die Zeit entsprechend eine Teilmenge der reellen bzw. der ganzen Zahlen ist. Für präzise Definitionen der Begriffe siehe etwa [JETSCHKE 1989](#).

Im Wesentlichen gibt es zwei Arten dynamischer Systeme:

„There are two main types of dynamical systems: differential equations and iterated maps (also known as difference equations).“ ([STROGATZ 2001](#), S. 5)

Phasenflüsse für kontinuierliche Systeme werden durch Differenzialgleichungssysteme beschrieben. Die Phasenflüsse zeitdiskreter dynamischer Systeme werden durch iterierte Abbildungen („iterated maps“) definiert. Sollen räumliche Abhängigkeiten des Zustands mit betrachtet werden, kommen partielle Differenzialgleichungen zum Einsatz.

So definierte dynamische Systeme sind eine mathematische Formalisierung eines deterministischen Prozesses ([JETSCHKE 1989](#), S. 24). Meist wird zusätzlich gefordert, dass dynamische Systeme *autonom* sind, d.h. der Phasenfluss ändert sich nicht mit der Zeit. Eine *Trajektorie* (auch *Orbit*) bezüglich eines Anfangszustands ist die zeitliche Folge aller Zustände entsprechend der Abbildung des Phasenflusses durch diesen Anfangszustand. Für deterministische dynamische Systeme verläuft durch jeden Punkt des Phasenraums genau eine Trajektorie ([JETSCHKE 1989](#), S. 25) und der gesamte Phasenraum ist mit Trajektorien gefüllt, da jeder Zustand als Anfangszustand dienen kann ([STROGATZ 2001](#), S. 7).

Ein *Phasenportrait* ist eine Darstellung charakteristischer Trajektorien eines dynamischen Systems, bei welchem die Richtung und Stabilitätseigenschaften gekennzeichnet sind. Ein Phasenportrait ist somit eine qualitative Charakterisierung möglicher Verhaltensarten eines dynamischen Systems, es zeigt „all the qualitatively different trajectories of the system“ ([STROGATZ 2001](#), S. 19). Der qualitativen Charakterisierung von dynamischen Systemen durch Phasenportraits liegt der Begriff der topologischen Äquivalenz zu Grunde: zwei

dynamische Systeme sind *topologisch äquivalent*, wenn es einen Homöomorphismus gibt, der Trajektorien des einen Systems auf Trajektorien des anderen Systems abbildet und umgekehrt (JETSCHKE 1989, S. 65). Phasenportraits von topologisch äquivalenten dynamischen Systemen sind „qualitatively similar“ (KUZNETSOV 1998, S. 39f). Bei der qualitativen Charakterisierung des Verhaltens von dynamischen Systemen durch Phasenportraits wird also von den konkreten Trajektorienverläufen abstrahiert und sich auf die Beschreibung essentieller Merkmale fokussiert.

### 5.1.2. Zeitliches Verhalten dynamischer Systeme

Die Charakterisierung zeitlichen Verhaltens erfolgt in der Theorie dynamischer Systeme durch eine Klassifikation von Trajektorien. Dabei werden Klassen von qualitativ ähnlichen Trajektorien identifiziert. Zusätzlich wird untersucht, wie sich das Verhalten gegenüber Störungen verhält, d.h. die Stabilität bestimmter Trajektorien wird betrachtet. Alternativ ist es auch möglich, die grafische Darstellung zeitlichen Verhaltens für eine phänomenologische Beschreibungen heranzuziehen. Darauf wird vor dem abschließenden [Beispiel 5.1](#) eingegangen.

#### Arten von Trajektorien

Welche Arten von Trajektorien treten in Phasenportraits auf? Die wichtigsten sind Fixpunkte und geschlossene Trajektorien sowie Trajektorien, die sich Fixpunkten und geschlossenen Trajektorien annähern:

„Bisher haben wir für das Langzeitverhalten einer Trajektorie (abgesehen von unbeschränktem Wachstum) drei Möglichkeiten gefunden: Entweder ist  $x(t)$  ein Fixpunkt oder eine geschlossene Trajektorie oder  $x(t)$  erreicht für  $t \rightarrow \infty$  asymptotisch einen Fixpunkt oder eine geschlossene Trajektorie.“ (JETSCHKE 1989, S. 54)

Ein *Fixpunkt* eines dynamischen Systems ist eine Trajektorie, die in einem einzelnen Zustand verharrt, eine *geschlossene Trajektorie* (auch *periodischer Orbit*, *Zyklus*) ist eine Trajektorie, in welcher sich die Zustände nach endlicher Zeit wiederholen.

Daneben gibt es noch Trajektorien, die weder Fixpunkt noch geschlossen sind:

„We can roughly classify all possible orbits in dynamical systems into fixed points, cycles, and ‚all others.‘“ (KUZNETSOV 1998, S. 10)

Zu diesen „anderen“ Trajektorien zählen unter anderem solche, die sich Fixpunkten und geschlossenen Trajektorien annähern (wie im obigen Zitat von JETSCHKE 1989).

#### Stabilität von Trajektorien

Einer der zentralen Begriffe der Theorie dynamischer Systeme ist die *Stabilität* von Fixpunkten und geschlossenen Trajektorien. Die Stabilität wird meist allgemein für invariante Mengen definiert. Eine invariante Menge  $A$  eines dynamischen Systems ist ein Teilbereich des Phasenraums, welcher von Trajektorien, die in  $A$  starten, nicht verlassen wird. Fixpunkte und geschlossene Trajektorien sind solche invariante Mengen. Generell ist jede einzelne

Trajektorie eine invariante Menge. Für die Stabilität einer invarianten Menge  $A$  wird das Verhalten von Trajektorien betrachtet, die nahe von  $A$  starten. Dies repräsentiert kleine Störungen des Zustands des dynamischen Systems.

Dabei gibt es zwei unabhängige Ansätze, die Stabilität invarianter Mengen zu definieren (die Bezeichnung der Stabilitätseigenschaften folgt KUZNETSOV 1998, für eine Diskussion der unterschiedlichen Bezeichnungsweisen siehe Abschnitt 6.2.3). Einerseits wird von einem stabilen System gefordert, dass Trajektorien, die nahe an  $A$  starten, in der näheren Umgebung von  $A$  verbleiben (für eine exakte mathematische Definition siehe z.B. KUZNETSOV 1998, S. 16). Diese wird oft „Ljapunow Stabilität“ genannt (nach dem russischen Mathematiker ALEXANDER MICHAJLOWITSCH LJAPUNOW). Andererseits wird das Langzeitverhalten von Trajektorien betrachtet, die nahe  $A$  starten. Nähern diese Trajektorien sich  $A$  asymptotisch, heißt  $A$  *asymptotisch stabil*. Oft treten beide Stabilitätseigenschaften gemeinsam auf. Eine invariante Menge, die sowohl Ljapunow-stabil als auch asymptotisch stabil ist, heißt *stabil*. Ist eine invariante Menge weder Ljapunow-stabil noch asymptotisch stabil, heißt sie *instabil*. Die beiden Stabilitätseigenschaften sind tatsächlich unabhängig. Es lassen sich sowohl Beispiele invarianter Mengen finden, die nur Ljapunow-stabil sind, als auch solche, die nur asymptotisch stabil sind (siehe STROGATZ 2001, S. 129 und KUZNETSOV 1998, S. 16f). Allerdings treten invariante Mengen, die nur asymptotisch stabil sind, nur sehr selten in dynamischen Systemen auf, die Beispiele wirken oft sehr konstruiert. Aus diesem Grund definieren nicht alle Autoren eine separate asymptotische Stabilität.

In der Theorie dynamischer Systeme gibt es noch zwei weitere wichtige Stabilitätsbegriffe:

- (1) Ein dynamisches System ist *strukturell stabil*, wenn sein Phasenportrait bei Parameteränderung qualitativ gleich bleibt (siehe Abschnitt 5.1.3 für eine ausführliche Darstellung).
- (2) Dynamische Systeme werden nach der Anzahl stabiler Zustände unterschieden: Ein dynamisches System heißt *monostabil*, falls es genau einen stabilen Fixpunkt besitzt, *bistabil* bei genau zwei stabilen Fixpunkten und *multistabil* bei mehr als zwei stabilen Fixpunkten. Die Stabilität von Trajektorien betrachtet also invariante Mengen, die Mono-, Bi- und Multistabilität das gesamte Phasenportrait und die strukturelle Stabilität den kombinierten Phasen-Parameter-Raum.

### Phänomenologische Beschreibung zeitlichen Verhaltens

Neben den Konzepten der Theorie dynamischer Systeme, die Phasenportraits, charakteristische Trajektorien und deren Eigenschaften betreffen, gibt es auch Beschreibungen zeitlichen Verhaltens, die sich direkt auf in Experimenten oder Simulationen beobachtete Veränderungen des Systemzustandes beziehen. Diese könnten als „phänomenologische Beschreibungen“ bezeichnet werden.

Typische Beispiele sind „Oszillation“ und „stationärer Zustand“. So bezieht sich etwa ein stationärer Zustand auf direkte Beobachtungen des zeitlichen Verhaltens, namentlich im konkreten Fall darauf, dass sich der Zustand nicht ändert. Dagegen betrifft die Definition von Fixpunkt eine Eigenschaft des gesamten dynamischen Systems:  $d\mathbf{x}/dt = 0$  ( $\mathbf{x}$  ist der Zustandsvektor). Die phänomenologischen Beschreibungen beziehen sich dabei häufig nur auf Unterräume des Phasenraums bzw. nur auf einzelne Dimensionen. So findet sich oft die Beobachtung, dass sich eine bestimmte Zustandsgröße im stationären Zustand befindet, d.h. nicht ändert. Beide Begriffe hängen offensichtlich zusammen. Dabei ist jedoch die Forderung eines Fixpunktes stärker: Befindet sich ein dynamisches System in einem Fixpunkt, ist ein

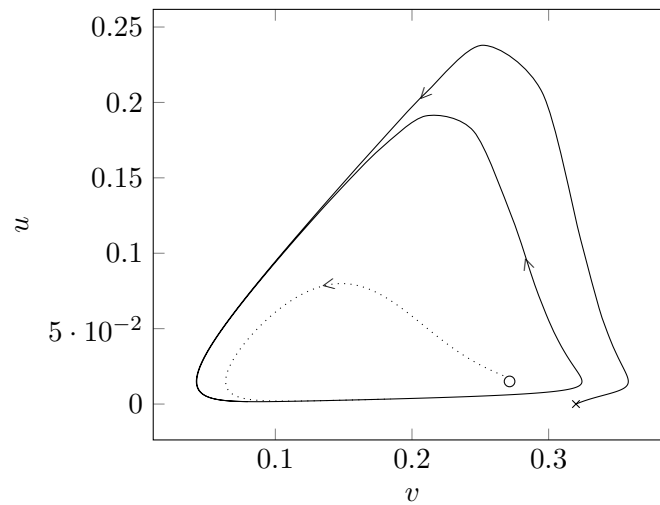


Abbildung 5.1.: Phasenportrait von **Modell 2** für  $k_6 = 1 \text{ min}^{-1}$  und die initialen Zustände  $(u, v) = (0.015, 0.275)$  (gepunktete Kurve) und  $(u, v) = (0, 0.32)$  (durchgezogene Kurve). Alle anderen Werte wurden wie in [Tabelle 1.2](#) gewählt. Der Kreis markiert den instabilen Fixpunkt  $(u^*, v^*) \approx (0.015, 0.2714)$  des Systems. Der initiale Zustand der gepunkteten Trajektorie liegt nahe an diesem Fixpunkt. Das Kreuz kennzeichnet den initialen Zustand, der auch in [Abbildung 5.2](#) verwendet wird. Die Simulation der Zeitreihen erfolgte in [COPASI](#) Version 4.10 (Build 55) für 25 min (gepunktete Kurve, Schrittweite 0.125 min) und 50 min (durchgezogene Kurve, Schrittweite 0.05 min).

stationäres Verhalten in allen Zustandsgrößen zu beobachten. Umgekehrt gilt dies nicht: aus der Beobachtung eines stationären Zustands einzelner Zustandsgrößen lässt sich noch nicht die Existenz eines Fixpunktes ableiten.

Es ist zu bemerken, dass die für die phänomenologischen Beschreibungen des zeitlichen Verhaltens eingesetzten Bezeichnungen meist physikalisch geprägt sind. Phänomenologische Beschreibungen werden auch für experimentelle Beobachtungen verwendet. Diese Tatsache soll durch das Wort „phänomenologisch“ hervorgehoben werden.

In diesem Zusammenhang von Interesse ist die Idee in [SCHAICH ET AL. 2001](#), welche starke Parallelen zur qualitativen Modellierung (vgl. [Abschnitt 5.2](#)) aufweist. Dabei werden Zeitreihen in Segmente eingeteilt, die jeweils dieselben Vorzeichen der ersten und zweiten Ableitung haben, d.h. denselben Anstieg und dieselbe Konvexität aufweisen. Jeder Segmenttyp wird durch ein Symbol repräsentiert. Es gibt vier verschiedene Kombinationen aus Anstieg und Konvexität sowie drei lineare Segmenttypen (zweite Ableitung ist null). Zeitreihen werden dann durch Folgen von Symbolen für die sieben Segmenttypen beschrieben. Dieser Ansatz erlaubt einen Übergang von einer quantitativen Beobachtung konkreten Verhaltens zu einer symbolischen qualitativen Repräsentation. Dieser Übergang kann leicht algorithmisch vollzogen werden. Dies könnte ein erster Schritt für ein Verfahren sein, welches aus numerischen Zeitreihen phänomenologische Beschreibungen generiert.

### Beispiel 5.1 (Modell 2 als dynamisches System)

In diesem Beispiel wird die hier beschriebene Theorie dynamischer Systeme anhand von **Modell 2** illustriert. Für die numerischen Simulationen wurde dabei [COPASI](#) (*C*Omplex

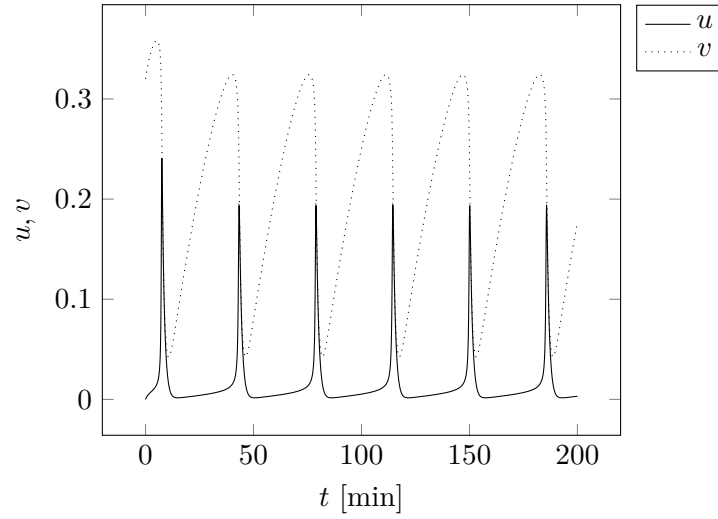


Abbildung 5.2.: Zeitreihen von Modell 2 für  $k_6 = 1 \text{ min}^{-1}$  und den initialen Zustand  $(u, v) = (0, 0.32)$  (markiert durch ein Kreuz in [Abbildung 5.1](#)). Alle anderen Werte wurden wie in [Tabelle 1.2](#) gewählt. Die Simulation der Zeitreihen erfolgte in [COPASI](#) Version 4.10 (Build 55) für 200 min und einer Schrittweite von 0.05 min.

*Pathway Simulator*, [HOOPS ET AL. 2006](#)) eingesetzt. [Abbildung 5.1](#) zeigt das Phasenportrait von Modell 2 für  $k_6 = 1 \text{ min}^{-1}$ , in welchem ein instabiler Fixpunkt (gekennzeichnet durch einen Kreis) und ein stabiler Grenzzyklus zu erkennen sind. Außerdem sind zwei Trajektorien gezeigt, die sich dem stabilen Grenzzyklus spiralförmig nähern: Eine Trajektorie startet in der Nähe des instabilen Fixpunktes, die andere außerhalb des Grenzzyklus (gekennzeichnet durch ein Kreuz). Ein Fixpunkt, von dem Trajektorien spiralförmig ausgehen, wird auch als *instabiler Strudel* bezeichnet ([JETSCHKE 1989](#), S. 40). Die Lage des Fixpunktes  $(u^*, v^*)$  lässt sich aus [Gleichung 1.17](#) und [Gleichung 1.18](#) für  $\frac{du}{dt} = \frac{dv}{dt} = 0$  ermitteln zu:

$$u^* = \frac{k_1[aa]}{k_6[CT]} \quad (5.1)$$

$$v^* = \frac{k_6 u^*}{k_4(\alpha + u^{*2})} + u^* . \quad (5.2)$$

Für die Parameter aus [Tabelle 1.2](#) ergibt sich  $u^* = 0.015$  und  $v^* \approx 0.2714$ . Die Stabilität des Fixpunktes kann anhand der Eigenwerte der Jacobi-Matrix  $A(u^*, v^*)$  des dynamischen Systems bestimmt werden. Die Jacobi-Matrix  $A(u, v)$  des dynamischen Systems Modell 2 an einem Punkt  $(u, v)$  des Phasenraums stellt eine Linearisierung des dynamischen Systems dar und besteht aus den partiellen Ableitungen des dynamischen Systems bezüglich aller abhängigen Variablen:

$$\begin{aligned} A(u, v) &= \begin{pmatrix} \frac{\partial f_u}{\partial u}(u, v) & \frac{\partial f_u}{\partial v}(u, v) \\ \frac{\partial f_v}{\partial u}(u, v) & \frac{\partial f_v}{\partial v}(u, v) \end{pmatrix} \\ &= \begin{pmatrix} k_4(-3u^2 + 2uv - \alpha) - k_6 & k_4(\alpha + u^2) \\ -k_6 & 0 \end{pmatrix} , \end{aligned} \quad (5.3)$$



wobei  $f_u(u, v) = k_4(v - u)(\alpha + u^2) - k_6u$  und  $f_v(u, v) = k_1[aa]/[CT] - k_6u$  die rechten Seiten von [Gleichung 1.15](#) und [Gleichung 1.18](#) darstellen. Für die Bewertung der Stabilität eines Fixpunktes  $(u^*, v^*)$  sind die Eigenwerte der Jacobi-Matrix an diesem Fixpunkt entscheidend. Die beiden komplex konjugierten Eigenwerte von  $A(u, v)$  sind allgemein:

$$\lambda_1(u, v) = \frac{\tau + \sqrt{\tau^2 - 4\Delta}}{2} \quad (5.4)$$

$$\lambda_2(u, v) = \frac{\tau - \sqrt{\tau^2 - 4\Delta}}{2} \quad (5.5)$$

$$\tau(u, v) = k_4(-3u^2 + 2uv - \alpha) - k_6 \quad (5.6)$$

$$\Delta(u, v) = k_4k_6(\alpha + u^2), \quad (5.7)$$

wobei  $\tau$  die Spur von  $A$  und  $\Delta$  die Determinante von  $A$  ist. Am Fixpunkt  $(u^*, v^*) = (0.015, 0.2714)$  ergeben sich mit  $\tau(u^*, v^*) \approx 0.3261 \text{ min}^{-1}$  und  $\Delta(u^*, v^*) = 0.0585 \text{ min}^{-2}$  die komplexen Eigenwerte (in  $\text{min}^{-1}$ ):

$$\lambda_1(u^*, v^*) \approx 0.1631 + 0.1786i \quad (5.8)$$

$$\lambda_2(u^*, v^*) \approx 0.1631 - 0.1786i. \quad (5.9)$$

Da die Realteile beider Eigenwerte  $\lambda_1(u^*, v^*), \lambda_2(u^*, v^*)$  am Fixpunkt  $(u^*, v^*)$  positiv sind, ist der Fixpunkt asymptotisch instabil. Für Details zur Methode der Linearisierung für die Bestimmung von Stabilitätseigenschaften und deren Grenzen siehe z.B. [JETSCHKE 1989](#) (S. 43ff) und [KUZNETSOV 1998](#) (S. 46ff). Die Berechnung der Jacobi-Matrix und der Eigenwerte kann auch numerisch erfolgen. So bietet z.B. [COPASI](#) eine Analyse von Fixpunkten an (Menüpunkt „Task/Steady-State“), die auch die Stabilitätseigenschaften ermittelt.

[Abbildung 5.2](#) zeigt die zeitlichen Integralkurven der Projektion des Phasenportraits auf  $u$  bzw.  $v$  für den initialen Zustand  $u_0 = 0, v_0 = 0.32$  (durchgezogene Kurve in [Abbildung 5.1](#), der initiale Zustand ist durch ein 'x' gekennzeichnet). Phänomenologisch lässt sich der Anfang beider Zeitverläufe als gedämpfte Oszillation bezeichnen, die bereits nach etwa einer Periode in eine periodische ungedämpfte Oszillation übergeht. ■

### 5.1.3. Qualitative Änderungen des Verhaltens: Bifurkationen

Die bisherige Betrachtung von Phasenportraits und charakteristischen Trajektorien bezogen sich immer auf eine konkrete Instanz eines dynamischen Systems, d.h. jeder Parameter hatte einen feststehenden Wert. Wie ändert sich ein Phasenportrait, wenn ein oder mehrere Parameter variiert werden? Bei kleinen Parameteränderungen gibt es meist keine qualitative Änderung des Phasenportraits, die Phasenportraits sind topologisch äquivalent. Eine Instanz eines dynamischen Systems heißt *strukturell stabil*, falls kleine Änderungen der Parameter zu topologisch äquivalenten Phasenportraits führen ([JETSCHKE 1989](#), S. 66). Es kann jedoch auch vorkommen, dass es an bestimmten Stellen des Parameterraums zu radikalen Veränderungen des qualitativen Systemverhaltens kommt, d.h. topologisch nicht äquivalente Phasenportraits bei kleinen Parameteränderungen auftreten. Solche qualitativen Veränderungen des Phasenportraits bei Variation von Parametern heißen *Bifurkationen* (auch *Verzweigungen*), die entsprechenden Stellen des Parameterraums *Bifurkationspunkte*

bzw. *kritische Werte* der betrachteten Parameter. Oft wird nur die Variation eines einzelnen, seltener von zwei Parametern untersucht und dabei alle anderen Parameter auf einem bestimmten Wert festgehalten. Die Anzahl der unabhängigen Parameter, die notwendig sind, einen bestimmten Bifurkationstyp zu definieren, wird als *Codimension* (kurz *codim*) bezeichnet (KUZNETSOV 1998, S. 63). Für die am häufigsten betrachteten Bifurkationen bezüglich eines Parameters (*codim*-1) wird häufig verkürzend von „am/unterhalb/oberhalb des Bifurkationspunktes“ gesprochen. Gemeint sind damit Parameterwerte, die gleich („am“), kleiner („unterhalb“) bzw. größer („oberhalb“) des kritischen Wertes für diesen Parameter liegen. Diese verkürzende Sprechweise wird auch im vorliegenden Text verwendet.

Es gibt unterschiedliche Typen von Bifurkationen, die sich in der Art und Weise der Phasenportraitänderungen ausdrücken. Die Bifurkationen eines dynamischen Systems können in einem *Bifurkationsdiagramm* dargestellt werden. Dabei unterteilen die Übergänge zwischen topologisch nicht äquivalenten Phasenportraits den Parameterraum. Die einzelnen Phasenportraits werden im Bifurkationsdiagramm skizziert oder lediglich benannt. Bei höherdimensionalen dynamischen Systemen werden nur Teilräume des Parameter- sowie des Phasenraums dargestellt. Auf eine detaillierte Darstellung verschiedener Bifurkationstypen wird an dieser Stelle verzichtet und auf die einschlägige Literatur verwiesen. Im folgenden [Beispiel 5.2](#) wird eine *codim*-1 Bifurkation beschrieben, die in *Modell 2* auftritt, wenn der Parameter  $k_6$  variiert wird.

### Beispiel 5.2 (Bifurkation von Modell 2)

In diesem Beispiel soll die von TYSON 1991 beschriebene Änderung des qualitativen Verhaltens von *Modell 2*, die bisher nur phänomenologisch anhand von *Modell 1* diskutiert wurde (siehe [Beispiel 2.6](#) und [Beispiel 2.7](#)), mittels der Theorie dynamischer Systeme analysiert und beschrieben werden.

[Abbildung 2.4](#) zeigt die numerische Berechnung der Bifurkationen von *Modell 1* bezüglich  $k_6$ , wobei alle anderen Parameter auf ihren Standardwerten ([Tabelle 1.2](#)) festgehalten werden. Hier sollen die beiden Bifurkationen durch die Methode der Linearisierung (siehe [Beispiel 5.1](#)) näher beleuchtet werden. Eine Bifurkation eines Fixpunktes, welche mit Hilfe der Linearisierung an diesem Fixpunkt analysiert werden kann, heißt auch *lokale Bifurkation* (KUZNETSOV 1998, S. 58). Bei einer solchen lokalen Bifurkation wechselt der Fixpunkt seine Stabilität. Beide Bifurkationen von *Modell 2* bezüglich  $k_6$  sind lokale Bifurkationen. Der Fixpunkt ändert bei diesen Bifurkationen neben seiner Stabilität auch seine Lage gemäß [Gleichung 5.1](#) und [Gleichung 5.2](#). In [Beispiel 5.1](#) wurde gezeigt, wie die Stabilität des Fixpunktes mit Hilfe der Eigenwerte der Jacobi-Matrix bestimmt werden kann. Die Stabilität des Fixpunktes wechselt, wenn die Realteile beider komplex konjugierten Eigenwerte gleichzeitig ihr Vorzeichen ändern. Eine rein analytische Untersuchung dieses Übergangs ist aufgrund der Gleichungsstruktur von [Gleichung 5.4](#) sehr schwierig. Allerdings kann sowohl  $\tau$  als auch der Ausdruck innerhalb der Wurzel ( $\tau^2 - 4\Delta$ ) in Polynome überführt und deren Nullstellen numerisch bestimmt werden (dies erfolgte mit MATLAB, MATrix LABoratory, <http://www.mathworks.de/>). Danach ergibt sich, dass  $\tau^2 - 4\Delta$  im Intervall zwischen den beiden Nullstellen  $k_6 = 0.19248 \text{ min}^{-1}$  und  $k_6 = 1.90093 \text{ min}^{-1}$  des entsprechenden Polynoms negativ ist. Die beiden Eigenwerte  $\lambda_1, \lambda_2$  sind in diesem Intervall komplex konjugiert mit Realteil  $\tau$ . Die Bifurkationspunkte ergeben sich in diesem Intervall aus den Nullstellen der Polynomdarstellung von  $\tau$ , die numerisch ermittelt wurden zu  $k_6^{(1)} = 0.36407 \text{ min}^{-1}$  und  $k_6^{(2)} = 1.46259 \text{ min}^{-1}$ . In [Abbildung 5.3](#) sind die Eigenwerte der

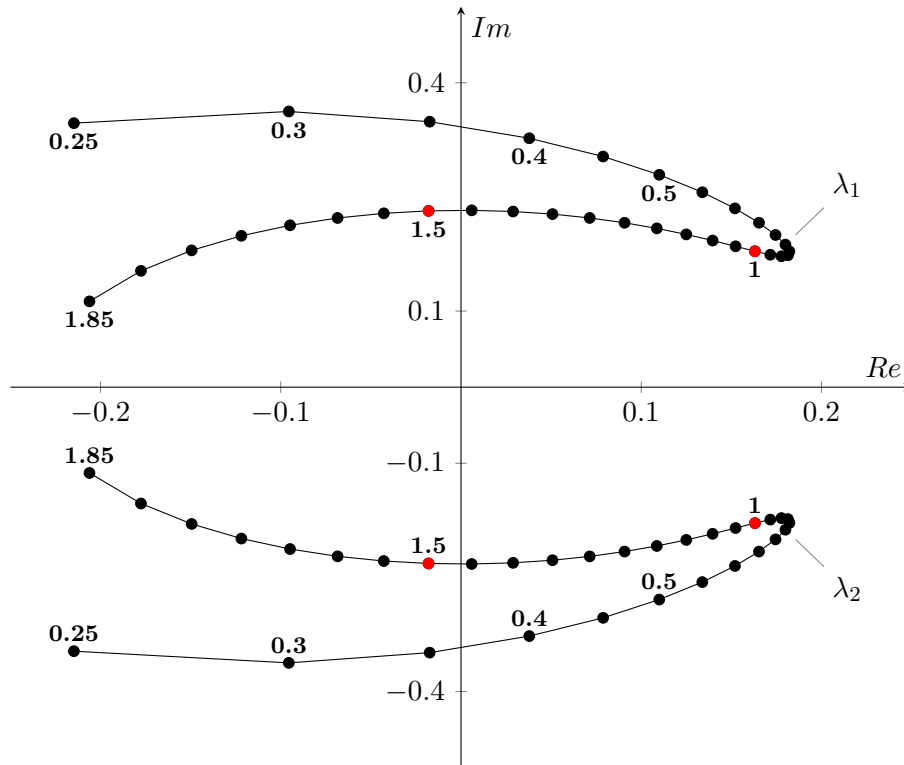


Abbildung 5.3.: Die komplex konjugierten Eigenwerte (in  $\text{min}^{-1}$ )  $\lambda_1(u^*, v^*)$  (oben) und  $\lambda_2(u^*, v^*)$  (unten) der Jacobi-Matrix von Modell 2 am Fixpunkt  $(u^*, v^*)$  für verschiedene Werte von  $k_6$  zwischen  $k_6 = 0.25 \text{ min}^{-1}$  und  $k_6 = 1.85 \text{ min}^{-1}$  in Schritten von  $0.05 \text{ min}^{-1}$ , dargestellt in der komplexen Zahlenebene. Alle anderen Werte wie in Tabelle 1.2. Einige Werte von  $k_6$  (in  $\text{min}^{-1}$ ) stehen fett gedruckt am jeweiligen Eigenwert. Die Eigenwerte für die Parameterbelegungen von Modell 2 in Abbildung 5.2 ( $k_6 = 1 \text{ min}^{-1}$ ) und Abbildung 5.4 ( $k_6 = 1.5 \text{ min}^{-1}$ ) sind rot hervorgehoben. Die einzelnen Eigenwerte sind für eine bessere Lesbarkeit miteinander durch Geraden verbunden. Die Eigenwerte wurden mit GNU Octave (<http://gnu.org/software/octave>) Version 3.6.4 berechnet.

Jacobi-Matrix von Modell 2 für verschiedene Werte von  $k_6$  dargestellt.

Unterhalb  $k_6^{(1)}$  sowie oberhalb  $k_6^{(2)}$  liegt ein stabiler Fixpunkt (Realteil der Eigenwerte negativ), dazwischen ein instabiler Fixpunkt (Realteil der Eigenwerte positiv). Eine Bifurkation, bei der zwei komplex konjugierte Eigenwerte die imaginäre Achse kreuzen, heißt *Hopf-Bifurkation* (JETSCHKE 1989, S. 96). Phänomenologisch ist eine Hopf-Bifurkation mit dem Auftreten oder dem Verschwinden eines Grenzzyklus verbunden, also mit Oszillation. Im Bereich zwischen den beiden Bifurkationspunkten weist Modell 2 einen stabilen Grenzzyklus auf und zeigt eine ungedämpfte Oszillation als Grenzverhalten (vgl. Abbildung 5.1 und Abbildung 5.2). Unterhalb und oberhalb der Bifurkationspunkte liegt ein stabiler Fixpunkt vor. Im Intervall  $k_6 = (0.19248, 1.90093) \text{ min}^{-1}$  handelt es sich dabei genauer um einen stabilen Strudel, da zwei komplex konjugierte Eigenwerte vorliegen (vgl. JETSCHKE 1989, S. 40). Alle Trajektorien nähern sich spiralförmig diesem Fixpunkt an, es ist eine gedämpfte Oszillation zu beobachten (Abbildung 5.4). ■

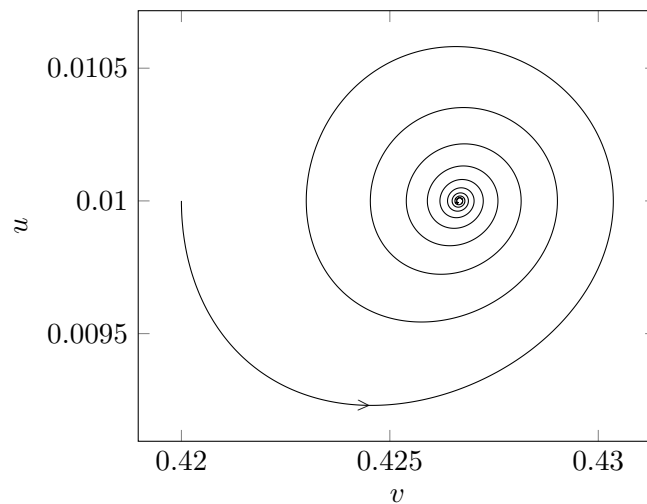


Abbildung 5.4.: Phasenportrait von Modell 2 für  $k_6 = 1.5 \text{ min}^{-1}$  und Anfangszustand  $u = 0.01$ ,  $v = 0.41$  (alle anderen Werte wie in [Tabelle 1.2](#)). Der Wert für  $k_6$  wurde nahe am Bifurkationspunkt gewählt, damit die gezeigte Trajektorie ausgeprägt spiralförmig auf den stabilen Fixpunkt zuläuft. Der Fixpunkt liegt gemäß [Gleichung 5.1](#) und [Gleichung 5.2](#) bei etwa  $(u, v) = (0.01, 0.4267)$ . Die Simulation der Zeitreihe erfolgte in [COPASI](#) Version 4.10 (Build 55) für 250 min und einer Schrittweite von 0.05 min.

## Fazit

Die Theorie dynamischer Systeme gestattet eine detaillierte qualitative Beschreibung des Verhaltens von Bio-Modellen. Die Dynamik eines Bio-Modells wird dabei durch Phasenportraits und deren charakteristische Trajektorien dargestellt. Bifurkationen erfassen qualitative Änderungen von Phasenportraits in Abhängigkeit von Parametern und somit die Diversifikation von Dynamik. Die Konzepte der Theorie dynamischer Systeme sind mathematisch definiert. Für eine formale Repräsentation der Dynamik von Bio-Modellen als Grundlage für eine Computerunterstützung fehlen jedoch sowohl eine formale begriffliche Basis (Ontologie) der Theorie als auch adäquate formale Ausdrucksmittel (formale Sprachen). In [Kapitel 6](#) wird eine entsprechende Ontologie entwickelt. [Abschnitt 5.2](#) und [Abschnitt 5.3](#) beschäftigen sich mit Techniken der KI zur formalen Beschreibung von Verhalten.

## 5.2. Qualitative Modellierung und Simulation

In diesem Abschnitt wird ein Überblick über die qualitative Modellierung und Simulation gegeben. Die qualitative Modellierung ist ein Forschungszweig der KI, der sich mit der symbolischen Repräsentation zeitlicher Aspekte der physikalischen Welt beschäftigt. Er wird oft auch als qualitative Physik oder *Qualitative Reasoning* bezeichnet.

Die KI hat sich schon früh mit qualitativen Beschreibungen physikalischer Zusammenhänge beschäftigt, etwa in der naiven Physik ([HAYES 1979](#)). Diese Tradition fortsetzend, beschäftigt sich die qualitative Modellierung und Simulation mit der formalen Repräsen-

tation kausaler Strukturen physikalisch-technischer Systeme und mit dem Schlussfolgern über das Verhalten der Systeme. Die qualitative Modellierung und Simulation ist aus drei klassischen Ansätzen hervorgegangen (DE JONG 2004, S. 94): Der komponentenbasierte Ansatz welcher in ENVISION (KLEER ET AL. 1984) realisiert wurde, die prozessbasierte QPT (*Qualitative Process Theory*, FORBUS 1984) und der constraintbasierte Ansatz, welcher in QSIM (KUIPERS 1986) verfolgt wird. Alle drei Ansätze wurden 1984 gemeinsam im Volume 24 der Zeitschrift „Artificial Intelligence“ vorgestellt (ein Nachdruck dieser Ausgabe ist BOBROW 1985).

Im Folgenden werden die wichtigsten Konzepte und Methoden der qualitativen Modellierung skizziert (Abschnitt 5.2.1) und gezeigt, wie qualitative Modelle in Simulationen verwendet werden (Abschnitt 5.2.2). Für Details zu den Konzepten und Methoden sei auf die einschlägige Literatur verwiesen. Gute Übersichtsartikel sind etwa FORBUS 2008, STRUSS 2000 und DE JONG 2004.

In dieser Arbeit geht es um quantitative Modelle biologischer Systeme. Der Fokus bei der Beschäftigung mit qualitativer Modellierung und Simulation in diesem Abschnitt liegt auf der qualitativen Beschreibung des Verhaltens (Abschnitt 5.2.3). Sollen die Techniken der qualitativen Modellierung und Simulation für die Beschreibung der Dynamik quantitativer Modelle genutzt werden, muss das quantitative Modell zunächst in ein entsprechendes qualitatives Modell überführt werden. Dies wird im Folgenden für Modell 1 beispielhaft gezeigt. Abschließend wird die Eignung qualitativer Modellierung und Simulation für die formale Repräsentation der Dynamik von Bio-Modellen bewertet.

### 5.2.1. Qualitative Modelle

In diesem Abschnitt wird gezeigt, wie dynamische Systeme qualitativ beschrieben werden können. Der entscheidende Unterschied zu quantitativen Modellen liegt in der Beschreibung des Systemzustandes. Während die Zustände in quantitativen Modellen durch Vektoren reeller Zahlen bzw. numerischer Approximationen reeller Zahlen beschrieben werden, erfolgt die Charakterisierung der Zustände in qualitativen Modellen durch diskrete Abstraktionen der reellwertigen Zustände. Als Träger des Zustandes muss zunächst jedoch eine Systemstruktur festgelegt werden.

#### Systemstruktur

Die Struktur eines qualitativen Modells wird ganz analog der Struktur von Bio-Modellen (Abschnitt 2.2.1) durch Objekte und Relationen zwischen diesen festgelegt. Den Objekten sind dabei Systemgrößen zugeordnet und die Relationen beschreiben Abhängigkeiten zwischen entsprechenden Systemgrößen. Nach STRUSS 2000 (440ff) gibt es zwei Sichtweisen auf Veränderungen in einem dynamischen System: die „prozessorientierte“ und die „komponentenorientierte“ Sichtweise: In der prozessorientierten Sicht werden Änderungen am Systemzustand durch aktive Prozesse, wie etwa chemische Reaktionen, bewirkt. Dagegen wird ein System in der komponentenorientierten Sicht als fest vorgegebene Menge von Komponenten aufgefasst, die über Verbindungen miteinander interagieren.

Bei der Modellkomposition wird die Struktur eines Systems nicht fest vorgegeben, sondern bezüglich einer gegebenen Aufgabe aus einer Bibliothek von Modellfragmenten konstruiert. Modellfragmente beschreiben dabei „some fundamental piece of the domain’s

physics, such as processes (e.g., liquid flows), devices (e.g., transistors), and objects (e.g., containers)“ (FALKENHAINER ET AL. 1991, S. 98), die Bibliothek der Modellfragmente wird auch *Domain Theory* genannt. Modellfragmente sind mit Vorbedingungen verknüpft, die für ihre Anwendbarkeit erfüllt sein müssen.

### Qualitative Zustände

Generell wird der Zustand durch verschiedene Systemgrößen charakterisiert. Jede Systemgröße hat einen eigenen Wertebereich, das Kreuzprodukt dieser Wertebereiche stellt den Zustandsraum des Systems dar. Ein Zustand ist dann ein Vektor der Werte für die einzelnen Systemgrößen, d.h. ein Element des Zustandsraums. Es gibt verschiedene Arten, Systemgrößen qualitativ darzustellen (vgl. FORBUS 2008, S. 366ff).

Die einfachste ist sicherlich eine boolesche Charakterisierung einer Systemgröße, etwa in „normal“ und „unnormal“. Drei mögliche Werte für eine Systemgröße bietet die „sign algebra“ (vgl. FORBUS 2008, S. 366): „negativ“ (–), „Null“ (0) oder „positiv“ (+). Eine Verallgemeinerung dieser beiden Ansätze ist die Charakterisierung einer Systemgröße durch eine endliche Menge symbolischer Werte, etwa „null“, „klein“, „mittelgroß“, „groß“, „riesig“.

Ein Nachteil dieser symbolischen Charakterisierung von Zuständen ist die Notwendigkeit, die Wertebereiche bereits bei der Modellierung festlegen zu müssen (FORBUS 2008, S. 367). Besser wäre hier eine Diskretisierung der Zustände zur Simulationszeit. Eine entsprechende Umsetzung erfolgt über sogenannte „*Limit Points*“ bzw. „*Landmarks*“, eine Spezialisierung der *Limit Points* (vgl. FORBUS 2008, S. 367f). Die *Limit Points* stellen kritische Werte des reellen Wertebereichs einer Zustandsgröße dar, an denen wesentliche Veränderungen stattfinden. Meist werden zusätzlich spezielle *Limit Points* für positiv und negativ unendlich festgelegt. Alle Werte zwischen zwei *Limit Points* werden dagegen als qualitativ äquivalent betrachtet. Der reelle Wertebereich wird also durch die *Limit Points* partitioniert. Eine Systemgröße wird dann entweder durch einen *Limit Point* angegeben oder durch ein offenes Intervall zwischen zwei aufeinanderfolgenden *Limit Points*. Nicht immer liegen die *Limit Points* als konkrete Zahlenwerte vor. Für die qualitative Simulation reicht es aus, die totale Ordnung der *Limit Points* zu kennen. Liegen numerische Werte für kritische Werte einer Zustandsgröße vor, kann die Zustandsgröße durch Intervalle qualitativ abstrahiert werden. Die Intervallarithmetik erlaubt Berechnungen mit Intervallen.

Eine weitere Möglichkeit der qualitativen Charakterisierung von Systemgrößen stellt die Angabe von Größenordnungsbeziehungen („order of magnitude“) dar (FORBUS 2008, S. 368f). Dabei werden Werte danach verglichen, wie „nahe“ sie beieinander liegen. Typische Relationen sind etwa „vernachlässigbar gegenüber“, „annähernd gleich“ und „von selber Größenordnung“ (vgl. STRUSS 2000, S. 447f). Für diese Größenordnungsbeziehungen können entsprechende Schlussregeln angegeben werden.

In der qualitativen Modellierung werden zur Charakterisierung eines Zustandes neben den qualitativen Werten für die Systemgrößen häufig auch deren erste Ableitungen nach der Zeit angegeben, wiederum qualitativ als „decreasing“ (fallend), „steady“ (gleichbleibend) bzw. „increasing“ (steigend). Es gibt auch Ansätze, die die zweite Ableitung berücksichtigen. So wird in SCHAICH ET AL. 2001 ein gemessener oder simulierter numerischer Zeitverlauf einer Systemgröße in zeitliche Abschnitte eingeteilt, in welchen die Vorzeichen der Systemgröße selbst sowie von deren erster und zweiter Ableitung gleichbleiben (siehe auch Abschnitt 5.1.2). Diese Abschnitte werden als *Episoden* bezeichnet, eine entsprechen-



de Folge von Episoden, die einen Zeitverlauf repräsentiert, als *Historie*. Es gibt sieben unterschiedliche Episoden (vgl. [SCHAICH ET AL. 2001](#), Fig. 3, S. 1376).

Aufgrund der Abstraktion von den quantitativen Systemgrößen kommt es erklärlicher Weise zu einem Informationsverlust. So kann es z.B. passieren, dass die Oszillation einer Systemgröße in einer zu groben Diskretisierung nicht mehr als solche erkennbar ist, sondern wie ein stationärer Zustand erscheint.

### Qualitative Beschreibung von Veränderung

Zur Modellierung eines dynamischen Systems gehört neben einer Festlegung der Struktur und des Zustandsraumes auch eine Beschreibung, wie sich der Zustand zeitlich ändert. Dies erfolgt durch Angabe von qualitativen Zusammenhängen zwischen Systemgrößen. Dabei werden sowohl direkte Zusammenhänge (*Proportionalitäten*) beschrieben als auch indirekte Zusammenhänge (*Einflüsse*).

Eine positive *Proportionalität* zwischen zwei Systemgrößen  $A$  und  $B$  liegt vor, wenn eine Erhöhung von  $A$  eine Erhöhung von  $B$  nach sich zieht, d.h. wenn es eine monoton in  $A$  wachsende Funktion  $f$  gibt, mit  $B = f(\dots, A, \dots)$ . Analog wird die negative Proportionalität definiert. Mit Hilfe der Proportionalitäten, der vier Grundrechenarten und einem qualitativen zeitlichen Ableitungsoperators können dynamische Systeme als **QDEs** (*Qualitative Differential Equation*) beschrieben werden, welche die Grundlage von **QSIM** sind. Eine Differentialgleichung kann durch eine **QDE** abstrahiert werden, indem sie in einzelne Gleichungen zerlegt wird, die jeweils nur eine Operation enthalten und „qualitative constraints“ genannt werden (siehe [KUIPERS 1994](#), S. 43ff). In [Beispiel 5.3](#) ist dieses Vorgehen für **Modell 1** demonstriert. Durch geeignete Bereichsbeschränkungen kann prinzipiell jede Differenzialgleichung durch eine **QDE** abstrahiert werden. Nicht-monotone Funktionen zwischen Systemgrößen werden dabei in monotone Segmente zerlegt ([FORBUS 2008](#), S. 371). Der Übergang zwischen diesen Segmenten wird in **QSIM** durch sogenannte „transitions“ realisiert ([KUIPERS 1994](#), S. 61). Jedes Segment entspricht damit einem eigenen Modellfragment mit entsprechenden Bereichsbeschränkungen als Vorbedingungen.

Für die Modellkomposition ist es erforderlich, dass Abhängigkeiten zwischen Systemgrößen lokal spezifiziert werden können. Insbesondere bei komplexen Systemen ist es schwierig oder unmöglich, lokale Abhängigkeiten ausschließlich über Proportionalitäten zu beschreiben, da eine Systemgröße meist durch mehrere andere Systemgrößen beeinflusst wird, oft mit gegenläufigen Effekten. In der qualitativen Modellierung werden lokale Abhängigkeiten durch sogenannte *Einflüsse* dargestellt: Ein Einfluss beschreibt einen Beitrag einer Systemgröße  $A$  zur Veränderung einer Systemgröße  $B$ , d.h. eine Monotonie der zeitlichen Ableitung von  $B$  bezüglich  $A$ . ([STRUSS 2000](#), S. 449)

#### Beispiel 5.3 (Modell 1 als QDE)

In diesem Beispiel wird [Gleichung 1.9](#) als **QDE** dargestellt. Zunächst wird dafür [Gleichung 1.9](#) ( $[M]/dt = [pM]F([M]) - k_5[\sim P][M] - k_6[M]$ ) in Einzelgleichungen zerlegt und entsprechende Hilfsvariablen  $v_0, \dots, v_7$  eingeführt (zum Vorgehen siehe [DE JONG 2004](#), S. 98):

$$\begin{aligned}\frac{d[M]}{dt} &= v_0 \\ v_0 &= v_1 + v_2\end{aligned}$$



$$\begin{aligned}
 v_1 &= [pM]v_3 \\
 v_3 &= F([M]) \\
 v_2 &= -v_4 \\
 v_4 &= v_5 + v_6 \\
 v_5 &= k_5 v_7 \\
 v_7 &= [\sim P][M] \\
 v_6 &= k_6 [M] .
 \end{aligned}$$

Diese Einzelgleichungen können nun direkt in qualitative Constraints überführt werden (Notation siehe [DE JONG 2004](#), Table 1, S. 98):

```

D/DT([M], v0)
ADD(v1, v2, v0)
MULT([pM], v3, v1)
M+([M], v3)
MINUS(v4, v2)
ADD(v5, v6, v4)
MULT(k5, v7, v5)
MULT([~P], [M], v7)
MULT(k6, [M], v6)
    
```

Dabei wird die Funktion  $F([M])$  durch eine positive Proportionalität  $M^+$  repräsentiert. Zusätzlich kann noch angegeben werden, dass sich bestimmte Größen nicht ändern:

```

CONSTANT(k5)
CONSTANT(k6)
CONSTANT([~P])
    
```

Die **QDE** für [Gleichung 1.9](#) besteht aus den angeführten qualitativen Constraints und einem geeigneten Zustandsraum für die Variablen. Da die Konzentrationen und kinetischen Konstanten positiv sind, ergibt sich folgender Zustandsraum, wobei hier lediglich die extremen *Landmarks*  $\infty$ , 0 und  $-\infty$  angegeben sind (vgl. [KUIPERS 1994](#), S. 40):

$$\begin{aligned}
 [M], [pM], [\sim P] &: 0 < \infty \\
 k_5, k_6 &: 0 < \infty \\
 v_0 &: -\infty < 0 < \infty \\
 v_1, v_3, v_4, v_5, v_6, v_7 &: 0 < \infty \\
 v_2 &: -\infty < 0 .
 \end{aligned}$$

■

### Ontologien für dynamische Vorgänge

In den Arbeiten zur qualitativen Physik spielen Ontologien eine große Rolle. Bereits bei [HAYES 1978](#) taucht dieser Begriff auf. Allerdings referiert insbesondere in den Arbeiten der 1980er und 1990er der Begriff „Ontologie“ mehr auf seine ursprüngliche philosophische Bedeutung als Theorie über die Konstituenten der realen (physikalischen) Welt und weniger auf die heute übliche informatische Lesart als Begriffssystem (vgl. [Abschnitt 3.3](#)). Ontologie

dient in der qualitativen Modellierung der Formalisierung der extrinsischen physikalischen Bedeutung der mathematischen Modelle und damit als Grundlage für wissensbasierte automatische Modellierung und Simulation:

„Modeling systems based on traditional mathematics tend to be informal about ontological issues. Informal decisions, based on experience with the world as well as professional expertise, are used to decide what entities should be included in a situation, what phenomena are relevant, and what simplifications are sensible. One goal of qualitative modeling is to make such tacit knowledge explicit, providing formalisms that can be used for automating (either fully or partially, depending on task) the modeling process itself.“ (FORBUS 2008, S. 371)

Nach FORBUS 2008 (S. 371ff) spielen drei Arten von Ontologien für die qualitative Modellierung eine Rolle: Komponentenontologien fassen Systeme als Netzwerke von Komponenten auf, Prozessontologien betrachten Prozesse als eigenständige Kategorie und „field ontologies“ betrachten räumliche Systemstrukturen.

Mitte der 1990er beginnt der Übergang von der philosophisch-theoretischen Ontologie im oben beschriebenen Sinne zu einer Realisierung von Ontologien als informatische Artefakte. Die Ontologie EngMath (*Engineering Mathematics Ontology*, GRUBER ET AL. 1994) formalisiert physikalische Größen, wie sie in mathematischen Modellen benutzt werden. PhysSys (*Ontology for Physical Systems*, BORST ET AL. 1997) vereint verschiedene Sichtweisen auf physikalische Systeme, wie etwa die Komponenten- und die Prozesssicht sowie die durch EngMath bereitgestellte Sicht auf physikalische Größen.

### 5.2.2. Verwendung qualitativer Modelle

Qualitative Modelle können für eine Vielzahl von Aufgaben verwendet werden. In STRUSS 2000 (S. 433ff) werden unterschiedliche Anwendungen qualitativer Modellierung angeführt. Es lassen sich zwei wichtige Anwendungsklassen ausmachen: Der Entwurf sowie die Diagnose technischer Systeme. Dabei ist im industriellen Bereich die modellbasierte Diagnose die vorherrschende Anwendung (STRUSS 2000, S. 456).

Sowohl für den modellbasierten Entwurf als auch für die modellbasierte Diagnose ist es notwendig, ein geeignetes qualitatives Modell für Verhaltensvorhersagen zu verwenden, d.h. das Modell zu simulieren (siehe unten). Bei der Modellkomposition werden für eine vorliegende Aufgabenstellung (des Entwurfs oder der Diagnose) aus den Modellfragmenten einer *Domain Theory* automatisch geeignete Modelle erzeugt (siehe Abschnitt 5.2.1). Klassische Arbeiten auf dem Gebiet der Modellkomposition sind FALKENHAINER ET AL. 1991 und IWASAKI ET AL. 1991. Die Verhaltensvorhersage der so zusammengesetzten Modelle werden mit den Zielvorgaben beim Entwurf bzw. mit den Beobachtungen bei der Diagnose verglichen. Bei Abweichungen wird das Modell systematisch verändert, es wird also eine gezielte Modellrevision vorgenommen. Zur Realisierung der Anwendungsaufgaben qualitativer Modellierung mit Hilfe von Modellkomposition, Verhaltensvorhersage und Modellrevision siehe STRUSS 2000.

### Qualitative Simulation

Eine qualitative Simulation berechnet für ein gegebenes qualitatives Modell, welches sich in einem bestimmten qualitativen Zustand befindet, mögliche Folgezustände. Dabei wird Zeit lediglich im Sinne von vorher/nachher aufgefasst, d.h. es gibt eine zeitliche partielle Ordnung der Zustände. Eine explizite Repräsentation von Zeit als reelle oder diskrete Größe gibt es dagegen nicht. Jeder Simulator für qualitative Modelle besitzt festgelegte Transitionsregeln, die zulässige Übergänge zwischen Zuständen festlegen (STRUSS 2000, S. 453f). Diese genügen meist gewissen Stetigkeitsbedingungen und berücksichtigen oft die zeitlichen Ableitungen der Systemgrößen (siehe Abschnitt 5.2.1). Die einzigen Änderungen, die beim Übergang von einem Zustand in den nächsten auftreten, sind die Überschreitung von *Limit Points* mindestens einer Systemgröße oder die Änderung der zeitlichen Ableitung mindestens einer Systemgröße. Dabei können gegebenenfalls neue *Limit Points* erzeugt werden. Die Momente des Zustandswechsels werden als *distinguirierte* Zeitpunkte bezeichnet (DE JONG 2004, S. 96).

Die Proportionalitäten und Einflüsse (siehe Abschnitt 5.2.1) beschränken die Folgezustände für einen gegebenen Zustand, d.h. sie stellen Constraints für Zustände dar. Das Finden eines Zustandsübergangs wird damit zu einem „constraint satisfaction problem“ (FORBUS 2008, S. 380). Aus diesem Grund können für die Simulation von qualitativen Modellen Constraintlöser eingesetzt werden (vgl. z.B. APT ET AL. 2005). Die klassischen Simulatoren, wie etwa QSIM, greifen auf sehr ähnliche Techniken zurück. Insbesondere kommen aufgrund der Art der Problembeschreibung deklarative Programmiersprachen zum Einsatz, wie etwa Lisp (ABELSON ET AL. 1996) und Prolog (NILSSON ET AL. 1995).

Für jeden Folgezustand wird diese Suche nach möglichen Folgezuständen wiederholt, bis jeweils ein stationärer Zustand (Zustand ohne Folgezustand) oder ein bereits generierter Zustand erreicht wird. In letzterem Fall handelt es sich um ein zyklisches Verhalten.

Es kann gezeigt werden, dass intervallbasierte Simulationsalgorithmen, wie etwa QSIM, vollständig in dem Sinne sind, dass für alle Trajektorien des durch das simulierte qualitative Modell abstrahierten Differenzialgleichungssystems qualitativ abstrahierte Zustandsfolgen erzeugt werden (STRUSS 2000, S. 447). Die Verfahren sind jedoch in der Regel nicht korrekt, d.h. es werden sogenannte „spurious behaviors“ generiert (siehe unten).

### Simulatoren

An dieser Stelle seien zwei Referenzimplementierungen für qualitative Simulatoren angeführt: Der klassische Simulator QSIM von KUIPERS, welcher in Lisp programmiert ist, sowie Garp3 (BREDEWEG ET AL. 2009), welcher in Prolog implementiert ist. Es sei angeführt, dass QSIM etwas in die Jahre gekommen ist: die letzte Version stammt von 1997. Dadurch wird eine heutige Benutzung stark erschwert. Aber auch Garp3 scheint nicht weiter entwickelt zu werden. Die letzte Aktualisierung stammt zwar aus dem Jahre 2012, das Forschungsprojekt, in dessen Rahmen Garp3 entwickelt wurde, endete aber mit dem 6. Forschungsrahmenprogramm der EU 2006. Garp3 ist zwar ohne besondere Schwierigkeiten lauffähig, die verwendeten Algorithmen sind jedoch gar nicht oder nur unzureichend dokumentiert.

## Qualitative Modellierung in der Biologie

Es gibt nur wenige Arbeiten, die qualitative Modellierung und Simulation für biologische Techniken einsetzen. [FORBUS 2008](#) (S. 384) führt drei wichtige Arbeiten an: [TRELEASE ET AL. 1996](#) modelliert Immunreaktionen mit einem auf der [QPT](#) basierenden Ansatz. Mit der qualitativen Simulation von Genregulationsnetzen beschäftigt sich [DE JONG ET AL. 2004](#). In [KING ET AL. 2005](#) wird [QSIM](#) benutzt, um metabolische Reaktionen zu identifizieren. In allen drei Arbeiten werden andere Anwendungen qualitativer Methoden auf biologische Systeme angeführt. Die meisten dieser wenigen Arbeiten stammen zudem aus den 1990er Jahren, als eine große Euphorie über die qualitativen Methoden herrschte.

Neben grundsätzlichen Problemen der qualitativen Modellierung und Simulation, die in [Abschnitt 5.2.3](#) besprochen werden, liegt ein Hauptgrund für den geringen Erfolg qualitativer Modellierung in der Biologie darin, dass insbesondere Entwurf und Diagnose von Systemen einen zur biologischen Forschung völlig verschiedenen Ausgangspunkt haben und somit die etablierten Methoden nur schwer anwendbar sind und auf geringe Akzeptanz stoßen: Beim Entwurf technischer Systeme sind die verwendbaren Komponenten meist vorgegeben und zumindest grob in ihrer Funktionalität und ihrem Verhalten beschrieben. Dabei ist die *Domain Theory* basierend auf der etablierten klassischen Physik bereits größtenteils vorhanden. Auch bei der modellbasierten Diagnose technischer Systeme liegt ein Modell des korrekt arbeitenden Systems bereits vor, eventuell ergänzt um entsprechende Modelle für Fehlverhalten. Dagegen geht es in der Biologie gerade darum, beobachtete Phänomene mechanistisch zu erklären und dafür performante und kompetente Modelle zu konstruieren (vgl. [Abschnitt 1.2](#)). Die Modellierung dient in der Biologie also der Entwicklung der *Domain Theory*. Darüber hinaus lassen sich methodische Unterschiede zwischen modellbasierter Diagnose technischer Systeme und der biologischen Modellierung ausmachen: Die modellbasierte Diagnose bedient sich einer abduktiven Schlussweise für das Finden der Ursachen beobachteten Fehlverhaltens. In der Systembiologie dagegen überwiegen induktive (bei der Modellierung von Beobachtungen) und deduktive (bei der Verwendung von Modellen in Simulationen) Schlussweisen.

### 5.2.3. Qualitatives Verhalten

Dieses Kapitel beschäftigt sich mit der qualitativen Beschreibung von Dynamik. Wie sehen entsprechende Beschreibungen der Dynamik eines qualitativen Modells aus? Für einen Anfangszustand liefert eine qualitative Simulation einen gerichteten Baum von Zuständen, wobei die Kanten die detektierten Zustandsübergänge repräsentieren und die Wurzel den Anfangszustand darstellt ([DE JONG 2004](#), S. 101). Dabei gibt es in [QSIM](#) zwei Zustandstypen, die auf jedem Pfad im Baum alternieren: Zustände, die zu einem bestimmten distinktierten Zeitpunkt gelten und Zustände die in einem Zeitintervall (zwischen zwei distinktierten Zeitpunkten) gelten (siehe auch [Abschnitt 5.2.2](#)).

Es ist auch möglich, alle möglichen Verhaltensweisen eines qualitativen Systems zu erzeugen. Der entstehende Graph wird häufig als „*Envisionment*“ ([STRUSS 2000](#), S. 455) bezeichnet. Ein Pfad in diesem Graphen stellt eine qualitative Abstraktion gleichartiger Trajektorien des dynamischen Systems dar. Der gesamte Graph abstrahiert also ein Phasenportrait (vgl. [Abschnitt 5.1](#)).

## Bifurkationen

Mit *Envisionments* können also Phasenportraits qualitativ charakterisiert werden. Bei Bio-Modellen ist man neben einer Charakterisierung einzelner Verhaltensweisen meist auch an qualitativen Verhaltensänderungen interessiert, die bei einer Variation der Randbedingungen auftreten. Sind solche Bifurkationen (vgl. [Abschnitt 5.1.3](#)) mit Hilfe qualitativer Modellierung darstellbar? Es gibt zwei Möglichkeiten, wie Bifurkationen in der qualitativen Modellierung auftreten können: Entweder explizit über separate Modellfragmente für verschiedene Phasenportraits unter- und oberhalb des Bifurkationspunktes oder implizit im Verhalten eines qualitativen Modells.

Die explizite Darstellung einer Bifurkation mittels qualitativer Modellierung erfordert zunächst, separate Systembeschreibungen für die qualitativ verschiedenen Phasenportraits unter- und oberhalb des Bifurkationspunktes zu erstellen. Betrachten wir folgendes dynamisches System, welches bei  $\mu = 0$  eine transkritische Bifurkation aufweist ([JETSCHKE 1989](#), S. 158f):

$$\dot{x} = \mu x - x^2. \quad (5.10)$$

$\mu x$  ist für  $\mu < 0$  monoton fallend in  $x$  und für  $\mu > 0$  monoton wachsend in  $x$ . Dieses System kann durch zwei unterschiedliche [QDEs](#) (siehe [Abschnitt 5.2.1](#)) qualitativ modelliert werden, einerseits mit einer positiven Proportionalität  $M^+(x)$ , andererseits mit einer negativen Proportionalität  $M^-(x)$ . Der Bifurkationsparameter taucht in diesen [QDEs](#) nicht mehr auf. Für die Modellkomposition könnten die beiden Fälle durch separate Modellfragmente repräsentiert werden, deren Vorbedingungen einen Vergleich von  $\mu$  mit dem Bifurkationspunkt 0 enthalten. Dadurch sind die beiden [QDEs](#) jeweils nur in einem Bereich der Randbedingungen gültig, in dem keine Bifurkation auftritt. Die Existenz und Lage von Bifurkationen sind jedoch insbesondere bei komplexen, nicht-linearen dynamischen Systemen in der Modellentwicklungsphase unbekannt.

Ein Bifurkationsparameter kann jedoch auch als Konstante ([CONSTANT](#), siehe [Beispiel 5.3](#)) in einer [QDE](#) repräsentiert sein. Es gibt zwei Fälle der Lage des Bifurkationspunktes bezüglich des Zustandsraums: Der Bifurkationspunkt ist (1) durch einen *Landmark* repräsentiert oder liegt (2) zwischen zwei *Landmarks*. Im Fall (1) zeigt sich je nach Anfangszustand des qualitativen Systems ein *Envisionment* entweder für unterhalb oder für oberhalb des Bifurkationspunktes. Der Fall (2) ist problematischer: Da die qualitativen Simulationsverfahren in der Regel vollständig sind ([Abschnitt 5.2.2](#)), müssen in einem solchen Fall die verschiedenen Verhaltensweisen unterhalb und oberhalb eines Bifurkationspunktes im selben *Envisionment* auftauchen. Der Zusammenhang zwischen der Variation des Parameters und der qualitativen Veränderung des Verhaltens ist dabei jedoch durch die Abstraktion zu einem qualitativen Modell völlig verloren gegangen. So kann etwa in [Beispiel 5.3](#) für  $k_6$  nur zwischen den beiden Werten 0 und  $\infty$  gewählt werden, die Bifurkation von [Modell 1](#) bezüglich  $k_6$  (siehe etwa [Beispiel 2.6](#)) ist damit für die qualitative Simulation „versteckt“.

## Probleme qualitativer Modellierung und Simulation

Die qualitative Modellierung scheint geeignet, dynamische Systeme und ihr Verhalten prägnant qualitativ zu erfassen. Es gibt jedoch zwei immanente Probleme der qualitativen Simulation, die diese Hoffnung zunichte machen (vgl. [DE JONG 2004](#)): Einerseits werden auch Zustandspfade erzeugt, die keine Abstraktion von reellwertigen Trajektorien sind.

Schwerwiegender als diese „spurious behaviors“ (STRUSS 2000, S. 454) ist andererseits die kombinatorische Explosion bei der Vorhersage von Folgezuständen. Jede Kombination von Änderungen einzelner Systemgrößen und deren zeitlicher Ableitungen wird in der Simulation als möglicher Folgezustand betrachtet. Zwar werden durch die Constraints einige dieser potentiellen Folgezustände als nicht zulässig erkannt. Die grobe Beschreibung eines qualitativen Modells bietet gerade bei großen und komplexen Systemen nicht genug Informationen, um die riesige Anzahl von potentiellen Folgezuständen auf ein erträgliches Maß zu reduzieren. Aus diesem Grund erlauben qualitative Modelle nur relativ schwache Aussagen über das Verhalten:

„In many situations, the QSIM predictions of the qualitative behaviors of a dynamic system are weaker than those obtained by techniques developed in dynamic systems theory.“ (DE JONG 2004, S. 111)

DE JONG 2004 schlägt als Lösungen für dieses Problem deshalb einerseits eine Rückkehr zu quantitativen Modellen und deren qualitative Analyse mit Hilfe der Theorie dynamischer Systeme (siehe Abschnitt 5.1) vor. Andererseits kann den Problemen mit semi-quantitativer Simulation begegnet werden.

## Fazit

Die qualitative Modellierung und Simulation weist einige Stärken auf, etwa die überschaubare Anzahl von Systemzuständen und die abstrakte Darstellung kausaler Abhängigkeiten zwischen Systemgrößen. Zusammen mit der durch die automatische Modellkomposition eingeführten Kontextabhängigkeit der Systembeschreibung eignet sich die qualitative Modellierung gerade für den komponentenbasierten Entwurf und die beobachtungs-basierte Diagnose technischer Artefakte.

Die qualitative Modellierung hat zwei oft angeführte Vorteile gegenüber der numerischen Simulation (DE JONG 2004, S. 93f): Zum einen müssen für die qualitative Simulation keine exakten numerischen Parameter und initialen Zustände vorliegen. Zum anderen ist das genaue zeitliche Verhalten oft nicht von primärem Interesse. Stattdessen,

„[...] we would like to know which classes of qualitatively-different behaviors are compatible with the structure of the system, given certain constraints on the values of the parameters and initial conditions.“ (DE JONG 2004, S. 93)

Erstaunlicherweise kommt der erste Vorteil im klassischen physikalisch-technischen Anwendungsbereich der qualitativen Simulation überhaupt nicht zum Tragen, da hier die Beschreibung der Systemkomponenten meist in hoher numerischer Präzision möglich ist. Dagegen sind genaue kinetische Parameter für die Modellierung biochemischer Systeme meist schwer zu erhalten (vgl. DE JONG 2004, S. 93).

Im Kontext der vorliegenden Arbeit ist gerade der zweite Vorteil erwähnenswert. Die durch qualitative Simulation erzeugten Zustandsgraphen erlauben eine qualitative Beschreibung der verschiedenen Verhaltensweisen eines dynamischen Systems.

Trotz all dieser wünschenswerten Eigenschaften scheint die qualitative Modellierung und Simulation zumindest in ihrer Reinform für die Systembiologie ungeeignet aus folgenden Gründen:



1. Modellfragmente gelten jeweils nur in einem bestimmten Bereich von Randbedingungen. Gerade bei den komplexen, nicht-linearen biologischen Systemen führt dies zu einer großen Zahl von Modellfragmenten. Die Darstellung ist also weniger kompakt als die eines entsprechenden quantitativen Bio-Modells.
2. Die oben angeführten Probleme führen eventuell zu einer unüberschaubaren Anzahl von Verhaltensweisen. Die Identifikation der typischen Verhaltensmuster des untersuchten Systems ist damit stark erschwert.
3. Durch die kombinatorische Explosion des Zustandsgraphen ist es sehr fragwürdig, ob ein Vorteil im Sinne des Berechnungsaufwands gegenüber numerischer Simulation besteht.
4. Das dynamische Verhalten eines Bio-Modells entsteht oft durch ein sehr fein abgestimmtes (bezüglich der Parameter) Wechselspiel der beteiligten Komponenten. Die Darstellung in qualitativen Modellen ist zu grob, um dem gerecht zu werden.
5. Qualitative Veränderungen des Verhaltens (Bifurkationen) müssen in qualitativen Modellen durch separate Modellfragmente repräsentiert werden, d.h. a priori bekannt sein. Für bestimmte dynamische Systeme sind Bifurkationen aus Sicht des qualitativen Modells nicht erkennbar.
6. Der Fokus der etablierten Verfahren der qualitativen Modellierung und Simulation auf Entwurf und Diagnose ist für die wissenschaftliche Methode der Systembiologie nicht angemessen.

Die Bedeutung der qualitativen Modellierung für die Beschreibung von Dynamik liegt in ihrer kompakten propositionalen Darstellung von Systemzuständen. Der folgende [Abschnitt 5.3](#) beschäftigt sich mit Temporalen Logiken, die es gestatten, zeitliche Beziehungen zwischen Systemzuständen auszudrücken.

### 5.3. Temporale Logiken

In diesem Abschnitt werden temporale Logiken vorgestellt. Temporale Logiken sind meist modale Erweiterungen der klassischen Logik, die es erlauben, Aussagen über zeitliche Folgen von Zuständen zu formulieren. Dafür ist es zunächst notwendig, ein dynamisches System in geeigneter Weise zu modellieren. [Abschnitt 5.3.1](#) beschäftigt sich mit sogenannten *Kripke-Strukturen*, die dynamische Systeme durch qualitative Zustände und Übergängen zwischen diesen darstellen. Bezüglich einer Kripke-Struktur können dann Aussagen über Zustände und Pfade im zugehörigen Zustandsübergangsgraph mit Hilfe von temporalen Logiken formuliert werden ([Abschnitt 5.3.2](#)). *Model Checking* ist eine Technik, in temporalen Logik formulierte Eigenschaften des Verhaltens von Kripke-Strukturen zu verifizieren ([Abschnitt 5.3.3](#)). Abschließend werden Ansätze vorgestellt, mit Hilfe des *Model Checkings* temporale Eigenschaften von biologischen Systemen zu untersuchen. Die folgende Darstellung sowie die Dreiteilung in Modellierung, Spezifikation von Eigenschaften und Verifikation folgen im Wesentlichen [CLARKE ET AL. 1999](#).



### 5.3.1. Modellierung dynamischer Systeme als Kripke-Struktur

Zunächst muss ein dynamisches System in eine Kripke-Struktur überführt werden, bevor deren temporale Eigenschaften charakterisiert werden können. Kripke-Strukturen sind somit qualitative Modelle für dynamische Systeme.

#### Reaktive und nebenläufige Systeme

Das *Model Checking* beschäftigt sich mit der Verifikation temporaler Eigenschaften reaktiver und nebenläufiger Systeme (siehe [CLARKE ET AL. 1999](#)). Ein dynamisches System ist *reaktiv*, wenn es mit der Umwelt interagiert. In *nebenläufigen* Systemen finden mehrere Aktivitäten parallel, d.h. unabhängig voneinander, statt. Einzelne dieser Aktivitäten können Reaktionen auf bestimmte Eingaben repräsentieren (vgl. [RUST 2005](#), S. 84f), so dass reaktive Systeme durch nebenläufige Systeme dargestellt werden können. [CLARKE ET AL. 1999](#) verwendet die beiden Begriffe „reaktives System“ und „nebenläufiges System“ synonym, ohne auf den Zusammenhang einzugehen. Im Folgenden spielt weder die Nebenläufigkeit noch die Reaktivität der Systeme eine Rolle, weshalb einfach von dynamischen Systemen gesprochen wird.

#### Kripke-Strukturen und Transitionssysteme

Dynamische Systeme werden im *Model Checking* durch sogenannte *Kripke-Strukturen* repräsentiert. Ein anderer, verwandter Ansatz sind Transitionssysteme. Während Kripke-Strukturen als zustandsbasiert angesehen werden können, sind Transitionssysteme ereignisbasiert. Diese Unterscheidung bezieht sich vorwiegend darauf, ob die Zustände (Kripke-Struktur) oder die Transitionen (Transitionssysteme) mit Eigenschaften bzw. Namen versehen sind (*Labeling*, siehe unten):

„The two types of structures are very similar and both can be seen as generalizations of state automata: in LTS's [Labelled Transition Systems – Anmerkung des Autors] transitions are labelled to describe the actions which cause a state change while in KS's [Kripke Structures – Anmerkung des Autors] states are labelled to describe how they are modified by the transitions.“ ([NICOLA ET AL. 1990](#), S. 407)

Oft werden beide Begriffe jedoch synonym gebraucht (z.B. [CLARKE ET AL. 1999](#), S. 27). [MONTEIRO ET AL. 2008](#) (S. i228) verwendet die Definition der Kripke-Struktur von [CLARKE ET AL. 1999](#) (S. 14) für „finite State Transition Systems“ (FSTSs). Transitionssysteme können in Kripke-Strukturen überführt werden ([CLARKE ET AL. 1999](#), S. 141). [BAIER ET AL. 2008](#) übernimmt in der Definition für Transitionssystem das *Labeling* der Zustände von den Kripke-Strukturen:

„We use transition systems with action names for the transitions (state changes) and atomic propositions for the states.“ ([BAIER ET AL. 2008](#), S. 20)

Im Folgenden werden Kripke-Strukturen als Transitionssysteme mit zusätzlichem *Labeling* der Zustände aufgefasst und nicht weiter auf die feinen Unterschiede zwischen beiden Begriffen eingegangen.

Eine *Kripke-Struktur*  $M$  über einer Menge  $AP$  atomarer Aussagen wird von CLARKE ET AL. 1999 (S. 14) als 4-Tupel  $M = (S, S_0, R, L)$  definiert, wobei  $S$  eine endliche Menge von Zuständen,  $S_0 \subseteq S$  die Menge der Startzustände,  $R \subseteq S \times S$  die Transitionsrelation und  $L : S \rightarrow 2^{AP}$  die *Labeling* Funktion ist, welche jedem Zustand in  $S$  eine Menge von atomaren Aussagen aus  $AP$  zuordnet, die in diesem Zustand wahr sind. Die Transitionsrelation ist linkstotal, d.h. für jeden Zustand gibt es einen Folgezustand. Um zu sichern, dass  $R$  linkstotal ist, kann für einen Zustand  $s \in S$ , der keinen ausgewiesenen Nachfolgezustand besitzt,  $R$  um  $R(s, s)$  erweitert werden (CLARKE ET AL. 1999, S. 15). Manchmal wird die Menge  $S_0$  der Startzustände auch weggelassen.

Diese Definition einer *Kripke-Struktur* stellt eine Kripke-Semantik für temporale Logiken bereit, woraus sich der Name „Kripke-Struktur“ ableitet. KRIPKE 1963 hat für die Modallogik eine Semantik eingeführt, die auf einer Menge möglicher Welten und einer Zugänglichkeitsrelation zwischen diesen Welten basiert. Die Kripke-Struktur definiert analog eine Menge von Zuständen (möglichen Welten) und Transitionen (Zugänglichkeitsrelation) für die Interpretation von temporallogischen Ausdrücken.

### Zustände und Zustandsübergangsgraphen

Die Zustände eines dynamischen Systems werden analog zu den Zuständen qualitativer Systeme (Abschnitt 5.2.1) als Vektoren von Systemgrößen definiert, die jeweils einen eigenen endlichen Wertebereich haben. Um solche Systemzustände mit einer Kripke-Struktur verbinden zu können, müssen geeignete atomare Aussagen in  $AP$  definiert sein. Typischerweise sind entsprechende atomare Aussagen von der Form  $v = d$ , wobei  $v$  eine Systemgröße ist und  $d$  ein Element ihres Wertebereichs (vgl. CLARKE ET AL. 1999, S. 14f). Eine atomare Aussage  $v = d$  kommt genau dann in  $L(s)$  vor, wenn im Zustand  $s \in S$  gilt, dass die Systemgröße  $v$  den Wert  $d$  hat. Auf diese Weise kann eine Belegung aller Systemgrößen in einem Zustand des dynamischen Systems in einen Zustand der Kripke-Struktur überführt werden.

Ein *Pfad*  $\pi$  in einer Kripke-Struktur  $M$  ausgehend von einem Zustand  $s \in S$  ist eine Zustandsfolge  $\pi = s_0 s_1 \dots$ , so dass  $s_0 = s$  gilt und  $R(s_i, s_{i+1})$  für alle  $i \geq 0$  gilt. Der gerichtete Graph  $(S, R)$  einer Kripke-Struktur  $M = (S, S_0, R, L)$  heißt *Zustandsübergangsgraph* von  $M$ .

### Überführung dynamischer Systeme in eine Kripke-Struktur

Kontinuierliche dynamische Systeme können in entsprechende Kripke-Strukturen überführt werden. MONTEIRO ET AL. 2008 schlägt ein Verfahren vor, bei dem Differenzialgleichungen zunächst durch stückweise lineare Differenzialgleichungen approximiert werden. Anschließend werden diese stückweise linearen Differenzialgleichungen durch diskrete Abstraktion in eine Kripke-Struktur überführt. Dabei repräsentiert jeder Zustand der Kripke-Struktur ein entsprechendes Hyperrechteck im Zustandsraum des kontinuierlichen Systems. Durch atomare Aussagen werden die Grenzen der Hyperrechtecke und die Änderungsrichtung von Systemgrößen repräsentiert. Diese Darstellung hat starke Parallelen zu der Zustandsbeschreibung qualitativer Systeme (vgl. Abschnitt 5.2.1).

In BIOCHAM (*Biochemical Abstract Machine*, CHABRIER-RIVIER ET AL. 2005) ist ein Verfahren implementiert, welches durch Regeln dargestellte Reaktionsgleichungen in eine

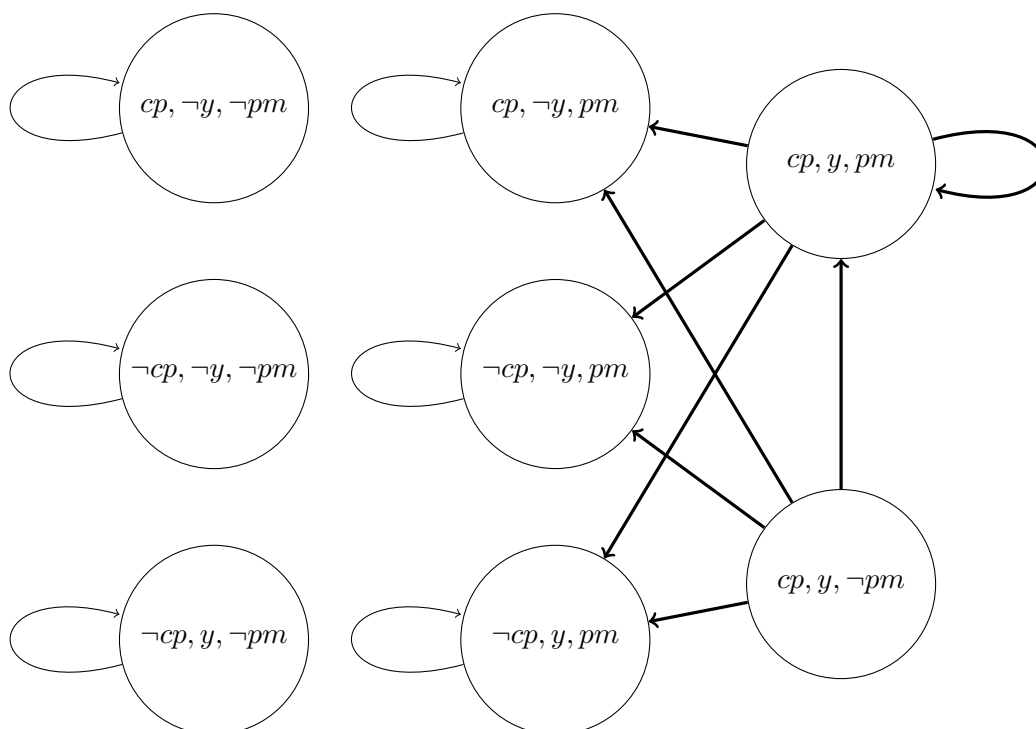


Abbildung 5.5.: Zustandsübergangsgraph für Reaktion 3 von Modell 1. Dicke Kanten repräsentieren ein Stattfinden der Reaktion, dünne Kanten repräsentieren Transitionen ohne Zustandsänderung. Für Details siehe [Beispiel 5.4](#).

Kripke-Struktur überführt. Dabei wird die Konzentration der beteiligten Substanzen durch boolesche Variablen repräsentiert (Substanz ist vorhanden/nicht vorhanden), der Zustand ist durch den Vektor aller solcher Variablen gegeben. Die Transitionsrelation ergibt sich als boolesche Abstraktion der Reaktionsregeln ([CHABRIER-RIVIER ET AL. 2005](#), S. 177). Für jede Reaktionsregel werden entsprechende Transitionen definiert, wobei alle möglichen Kombinationen der Zustände bezüglich der Reaktanten betrachtet werden (ein Reaktant kann durch die Reaktion verbraucht werden oder nicht). Im folgenden [Beispiel 5.4](#) wird eine entsprechende Kripke-Struktur für eine Reaktion aus Modell 1 präsentiert.

#### Beispiel 5.4 (Reaktion 3 von Modell 1 als Kripke-Struktur)

Die Überführung von Modell 1 in eine Kripke-Struktur soll an Reaktion 3 ([Abbildung 1.3](#)) demonstriert werden. Die atomaren Aussagen bezüglich der drei beteiligten chemischen Spezies  $CP$ ,  $Y$  und  $pM$  ergeben sich zu:

$$AP = \{c =_{df} [CP] > 0, y =_{df} [Y] > 0, p =_{df} [pM] > 0\} ,$$

wobei  $c$ ,  $y$  und  $p$  boolesche Variablen sind, die die entsprechenden Aussagen repräsentieren. Die Kripke-Struktur über  $AP$  für Reaktion 3 kann wie folgt definiert werden. Die Menge der Zustände  $S$  sind alle  $2^3$  Belegungen für die drei booleschen Variablen:

$$S = \{(0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 1, 0), (1, 0, 0), (1, 0, 1), (1, 1, 1), (1, 1, 0)\} .$$

Die Transitionen ergeben sich entsprechend der Reaktionsgleichung ( $CP + Y \rightarrow pM$ ):

$$\begin{aligned}
R = \{ & ((0, 0, 0), (0, 0, 0)), \\
& ((0, 0, 1), (0, 0, 1)), \\
& ((0, 1, 1), (0, 1, 1)), \\
& ((0, 1, 0), (0, 1, 0)), \\
& ((1, 0, 0), (1, 0, 0)), \\
& ((1, 0, 1), (1, 0, 1)), \\
& ((1, 1, 1), (1, 1, 1)), \\
& ((1, 1, 1), (0, 1, 1)), \\
& ((1, 1, 1), (1, 0, 1)), \\
& ((1, 1, 1), (0, 0, 1)), \\
& ((1, 1, 0), (1, 1, 1)), \\
& ((1, 1, 0), (0, 1, 1)), \\
& ((1, 1, 0), (1, 0, 1)), \\
& ((1, 1, 0), (0, 0, 1)) \} .
\end{aligned} \tag{5.11}$$

Die Reaktion findet statt, falls die beiden Reaktanten vorhanden sind (die unteren 8 Zeilen von [Gleichung 5.11](#)), d.h. für alle Zustände  $s = (1, 1, x)$ , mit  $x \in 0, 1$ . Für alle anderen Zustände findet die Reaktion nicht statt und der Zustand bleibt unverändert (die ersten sechs Zeilen von [Gleichung 5.11](#)). Findet die Reaktion statt, kann jeder Reaktant entweder verbraucht werden oder nicht. Dadurch ergeben sich jeweils vier Folgezustände. Das *Labeling*  $L$  ordnet einfach jedem Zustand die entsprechenden atomaren Aussagen zu:

$$\begin{aligned}
L((0, 0, 0)) &= \{\neg c, \neg y, \neg p\}, \\
&\vdots \\
L((1, 1, 1)) &= \{c, y, p\} .
\end{aligned}$$

Die Zustände und Zustandsübergänge sind als Zustandsübergangsgraph in [Abbildung 5.5](#) dargestellt. ■

An [Beispiel 5.4](#) werden zwei Probleme der Darstellung dynamischer Systeme als Kripke-Strukturen deutlich. Einerseits wächst die Anzahl der Systemzustände mit der Anzahl der Systemgrößen exponentiell: Eine Kripke-Struktur mit  $n$  binären Systemgrößen hat  $2^n$  verschiedene Zustände. Dieses Problem wird auch als „state explosion problem“ bezeichnet und stellt eine der größten Herausforderungen für das *Model Checking* dar ([CLARKE ET AL. 1999](#), S. 1). Dadurch ist auch die Anzahl möglicher Transitionen exponentiell in der Anzahl der Systemgrößen (da  $R$  linkstotal ist, gibt es mindestens genauso viele Transitionen wie Zustände). Somit ist eine explizite Aufzählung der Transitionen wie in [Beispiel 5.4](#) bei größeren Systemen offensichtlich ungeeignet. Die Kripke-Struktur muss implizit spezifiziert werden, etwa mit Regeln wie bei [BIOCHAM](#) (siehe [Abbildung 5.6](#)). Andererseits ist eine boolesche Repräsentation der Zustände zu grob, um die quantitativen Folgen von Reaktionen erfassen zu können. Damit die Darstellung von Reaktionen als Transitionen eine korrekte Abstraktion des kontinuierlichen Systems darstellt, müssen

alle möglichen qualitativen Folgezustände angegeben werden. In [Beispiel 5.4](#) führt das jeweils zu einer vierfachen Verzweigung im Verhalten für jedes Stattfinden der Reaktion: Jeder der beiden Reaktanten wird durch die Reaktion verbraucht oder nicht. Hier liegt ein analoges Problem wie das der „spurious behaviors“ in der qualitativen Modellierung vor (vgl. [Abschnitt 5.2.3](#)).

### 5.3.2. Spezifikation temporaler Eigenschaften

In diesem Abschnitt werden temporale Logiken betrachtet, die es gestatten, zeitliche Eigenschaften einer Kripke-Struktur auszudrücken. Die meisten temporalen Logiken sind Modallogiken, bei denen die üblichen Modaloperatoren  $\Box$  (notwendig) und  $\Diamond$  (möglich) für eine beliebige Aussage  $p$  zeitlich gedeutet werden als:

$$\begin{aligned}\Box p: & p \text{ gilt immer ,} \\ \Diamond p: & p \text{ gilt manchmal .}\end{aligned}$$

Im Folgenden werden Realisierungen dieser temporalen Modallogik vorgestellt, wie sie im *Model Checking* zur Anwendung kommen. Die Darstellung folgt [CLARKE ET AL. 1999](#) (S. 27ff).

#### Temporale Operatoren und Pfadquantoren

Temporallogische Formeln, wie sie unten eingeführt werden, beziehen sich nicht direkt auf die Kripke-Strukturen, sondern auf sogenannte „*Computation Trees*“. Ein *Computation Tree* für eine Kripke-Struktur und einen initialen Zustand ist der Baum von Zuständen und gerichteten Kanten, die jeweils eine Anwendung einer Transition darstellen. Es handelt sich somit um eine „abgewinkelte“ Darstellung des Zustandsübergangsgraphen. Da die Transitionsrelation linkstotal ist (siehe [Abschnitt 5.3.1](#)), ist der *Computation Tree* unendlich: Jeder Zustand (Knoten des Baums) hat einen Folgezustand, repräsentiert durch eine Kante.

Bezüglich eines *Computation Trees* können die beiden temporallogischen *Pfadquantoren* definiert werden, die eine Realisierung der oben angeführten zeitlichen Modaloperatoren darstellen:

**A** („for all“): für alle Pfade im *Computation Tree*,

**E** („exists“): für einige Pfade im *Computation Tree*.

Mit den Pfadoperatoren können temporallogische Aussagen bezüglich der Wurzel des *Computation Tree* oder für einen betrachteten Knoten formuliert werden. Zusätzlich zu den Pfadquantoren werden die folgenden *temporalen Operatoren* definiert, die Aussagen über Eigenschaften bezüglich eines einzelnen Pfades im *Computation Tree* erlauben:

**X** („next time“): die Eigenschaft gilt im nächsten Zustand auf dem Pfad,

**F** („in the future“): die Eigenschaft gilt irgendwann auf dem Pfad,

**G** („globally“): die Eigenschaft gilt auf dem gesamten Pfad,

**U** („until“): es gibt einen Knoten auf dem Pfad, an dem eine zweite Eigenschaft gilt. Die erste Eigenschaft gilt auf dem Pfad bis zu diesem Knoten.

**R** („release“): eine zweite Eigenschaft gilt entlang des Pfades bis zu einem Knoten, in dem die erste Eigenschaft gilt.

### Temporallogische Formeln

Mit den Pfadquantoren und temporalen Operatoren kann die temporale Logik **CTL\*** (*Computation Tree Logic*) über einer Menge  $AP$  von atomaren Aussagen definiert werden (vgl. CLARKE ET AL. 1999, S. 28f). In **CTL\*** gibt es zwei Arten von Formeln: *Zustandsformeln* (Aussagen über einen bestimmten Zustand) und *Pfadformeln* (Aussagen über einen bestimmten Pfad). Zustandsformeln sind wie folgt induktiv definiert:

- Jede atomare Aussage ist eine Zustandsformel.
- Sind  $f$  und  $g$  Zustandsformeln, so auch  $\neg f$ ,  $f \wedge g$  und  $f \vee g$ .
- Ist  $f$  eine Pfadformel, so sind  $\mathbf{E}f$  und  $\mathbf{A}f$  Zustandsformeln.

Die Pfadformeln werden wie folgt induktiv definiert:

- Jede Zustandsformel ist eine Pfadformel.
- Sind  $f$  und  $g$  Pfadformeln, so auch  $\neg f$ ,  $f \wedge g$ ,  $f \vee g$ ,  $\mathbf{X}f$ ,  $\mathbf{F}f$ ,  $\mathbf{G}f$ ,  $f\mathbf{U}g$  und  $f\mathbf{R}g$ .

Die Zustandsformeln werden kurz auch als **CTL\***-Formeln bezeichnet.

In **CTL\***-Formeln können Zustandsformeln und Pfadformeln also beliebig miteinander verschränkt sein. Es gibt auch temporale Logiken, bei denen dies nicht erlaubt ist (siehe unten). Die Semantik von **CTL\***-Formeln dürfte intuitiv klar sein. Für Zustandformeln und Pfadformeln lassen sich entsprechende Folgerungsbegriffe (vgl. Abschnitt 3.1.1) definieren:  $M, s \models f$  bedeutet, dass die Zustandsformel  $f$  im Zustand  $s$  bezüglich der Kripke-Struktur  $M$  wahr ist. Für eine Pfadformel  $f$  bedeutet  $M, \pi \models f$ , dass  $f$  entlang des Pfades  $\pi$  bezüglich der Kripke-Struktur  $M$  gilt. Die Relation  $\models$  wird induktiv über der Struktur temporallogischer Aussagen bezüglich einer gegebenen Kripke-Struktur definiert (siehe CLARKE ET AL. 1999, S. 29). Dabei dient das *Labeling*  $L$  der Kripke-Struktur als Interpretation atomarer Formeln.

### Einschränkungen der **CTL\***

Es gibt zwei wichtige Untersprachen von **CTL\***: **LTL** (*Linear Temporal Logic*) und **CTL** (*Computation Tree Logic*, eingeschränkte Version). Diese entstehen durch Beschränkung der Verschränkbarkeit von Zustands- und Pfadformeln.

In **LTL** sind Pfadformeln wie folgt beschränkt:

- Jede atomare Aussage ist eine Pfadformel.
- Sind  $f$  und  $g$  Pfadformeln, so auch  $\neg f$ ,  $f \wedge g$ ,  $f \vee g$ ,  $\mathbf{X}f$ ,  $\mathbf{F}f$ ,  $\mathbf{G}f$ ,  $f\mathbf{U}g$  und  $f\mathbf{R}g$ .

**LTL**-Formeln haben die Form  $\mathbf{A}f$ , wobei  $f$  eine Pfadformel ist. Da Formeln in **LTL** stets über alle Pfade quantifiziert sind, spielen mögliche Verzweigungen im *Computation Tree* keine Rolle mehr. Aus Sicht von **LTL** besteht der *Computation Tree* nur aus einer *linearen* Folge von Zuständen, die eine Vereinigung aller alternativen Pfade darstellen. Damit kann

jedoch nicht mehr formuliert werden, dass eine Eigenschaft auf manchen Pfaden gilt (BAIER ET AL. 2008, S. 314).

CTL ist eine Beschränkung der CTL\*, die Verzweigungen im *Computation Tree* berücksichtigt, die also auf einer „branching time“ (CLARKE ET AL. 1999, S. 30) basiert. Jedem temporalen Operator muss in einer CTL-Formel unmittelbar ein Pfadquantor vorausgehen. Um dies zu gewährleisten, sind Pfadformeln in CTL wie folgt beschränkt:

- Sind  $f$  und  $g$  Pfadformeln, so auch  $\mathbf{X}f$ ,  $\mathbf{F}f$ ,  $\mathbf{G}f$ ,  $f\mathbf{U}g$  und  $f\mathbf{R}g$ .

#### Beispiel 5.5 (Temporale Eigenschaften von Modell 1)

Für Modell 1 können z.B. folgende temporale Eigenschaften in CTL\* bezüglich aktivem MPF (*Maturation Promoting Factor*) formuliert werden, wobei  $m =_{df} [M] > 0$ . Die Übersetzung der Aussagen folgt dabei CHABRIER-RIVIER ET AL. 2004 (S. 38f).

- **Erreichbarkeit:** Ist, ausgehend von einem gegebenen Zustand MPF erreichbar, d.h. gibt es (E) eine Reihe von Reaktionen (Transitionen), die (irgendwann, F) zur Produktion von MPF führen?

$$\mathbf{EF}m$$

- **Stationärer Zustand:** Ist ein stationärer Zustand bezüglich MPF erreichbar, d.h. gibt es einen Pfad (E), auf dem ein Zustand erreicht wird (F), so dass auf allen von diesem Zustand startenden Pfaden (A) MPF immer (G) vorhanden ist?

$$\mathbf{EF}(\mathbf{AG}m)$$

Diese Forderung kann dahingehend abgeschwächt werden, dass der stationäre Zustand nur auf einem Pfad (E) gelten muss. Das System kann in diesem Fall dem stationären Zustand „entkommen“ (vgl. CHABRIER-RIVIER ET AL. 2004, S. 38).

$$\mathbf{EF}(\mathbf{EG}m)$$

- **Oszillation:** Kann sich der Zustand von MPF zyklisch ändern, d.h. gibt es einen Pfad (E), auf welchem stets (G) ein Zustandswechsel von MPF erfolgt? Der Zustandswechsel (von  $m$  nach  $\neg m$  oder umgekehrt) muss dabei auf mindestens einem Pfad (innere E) irgendwann (F) erfolgen.

$$\mathbf{EG}((m \rightarrow \mathbf{EF}\neg m) \wedge (\neg m \rightarrow \mathbf{EF}m))$$

■

#### 5.3.3. Verifikation temporaler Eigenschaften

In diesem Abschnitt soll das *Model Checking* vorgestellt werden, mit dessen Hilfe temporallogische Aussagen bezüglich einer Kripke-Struktur verifiziert werden können. Auf die verschiedenen Verfahren des *Model Checkings* wird dabei nicht eingegangen, stattdessen geht es um eine Vermittlung der grundsätzlichen Idee. Für Details sei auf die einschlägige Literatur verwiesen, etwa CLARKE ET AL. 1999.



### Model Checking

Allgemein soll durch *Model Checking* eine gegebene Kripke-Struktur  $M$  bezüglich einer Spezifikation, d.h. einer temporallogischen Formel  $f$ , *verifiziert* werden. Dafür wird folgendes Problem gelöst: Für welche Zustände  $s \in S$  erfüllt  $M$  die gegebene Spezifikation  $f$ , d.h.

$$M, s \models f ?$$

*Model Checking* stellt also eine Form der Abduktion dar, d.h. es wird von einer Beobachtung ( $f$ ) auf eine Erklärung ( $s$ ) bezüglich eines gegebenen Erklärungsrahmens ( $M$ ) geschlossen. Sind initiale Zustände  $S_0$  gegeben, so erfüllt eine Kripke-Struktur  $M$  eine durch  $f$  gegebene Spezifikation, falls für alle initialen Zustände  $s_i \in S_0$  gilt:  $M, s \models f$  (CLARKE ET AL. 1999, S. 35).

Auf konkrete Verfahren, Lösungen für dieses Problem zu finden, soll hier nicht eingegangen werden. Grundsätzlich gibt es zwei Arten von Verfahren: eine gegebene temporallogische Formel (Spezifikation) wird entweder durch Exploration des *Computation Trees* verifiziert (explizites *Model Checking*) oder durch Manipulation boolescher Formeln (symbolisches *Model Checking*). Im letzteren Fall werden dafür CTL-Formeln in binäre Entscheidungsdiagramme überführt und LTL-Formeln in prädikatenlogische Ausdrücke, die auf Erfüllbarkeit getestet werden. Für Details siehe z.B. CLARKE ET AL. 1999.

Im Allgemeinen ist *Model Checking* sehr teuer: Für CTL\* ist *Model Checking* PSPACE-vollständig in der Länge der Formel CLARKE ET AL. 1999 (S. 5f). Dies liegt insbesondere in der Größe des Zustandsübergangsgraphen begründet (vgl. „state explosion problem“ in Abschnitt 5.3.1).

### Anwendungen in der Systembiologie

Es gibt einige Versuche, temporale Logiken und *Model Checking* für die Charakterisierung des Verhaltens biologischer Systeme einzusetzen. Dafür müssen die dynamischen Modelle zunächst in eine geeignete Kripke-Struktur übertragen werden (siehe Abschnitt 5.3.1). Anschließend können Anfragen an die Kripke-Struktur in temporaler Logik formuliert und durch *Model Checking* verifiziert werden.

MONTEIRO ET AL. 2008 untersuchen typische biologische Anfragen, die an Bio-Modelle gestellt werden und entwickeln temporallogische Muster für diese Anfragen. Ein solches Muster ist etwa: „It is possible for a state  $\Phi$  to occur“ (MONTEIRO ET AL. 2008, S. i228). Die Muster werden automatisch in temporallogische Aussagen übersetzt und können durch *Model Checking* verifiziert werden.

In MONTEIRO ET AL. 2008 (S. i227) werden frühere Arbeiten angeführt, die *Model Checking* in der Systembiologie anwenden. Unter diesen befinden sich CHABRIER-RIVIER ET AL. 2004, in welchem ebenfalls typische Anfragen an Bio-Modelle in temporaler Logik formuliert werden (siehe Beispiel 5.5).

Mit BIOCHAM (*Biochemical Abstract Machine*, CHABRIER-RIVIER ET AL. 2005) lassen sich Bio-Modelle als Kripke-Strukturen modellieren und deren temporale Eigenschaften verifizieren. Zusätzlich zur Verifikation können auch Erklärungen für die Gültigkeit temporaler Aussagen in Form entsprechender Pfade der Kripke-Struktur generiert werden. Außerdem ist es möglich, die Gültigkeit temporallogischer Aussagen in Abhängigkeit von Parametern zu untersuchen. Es ist in BIOCHAM möglich, SBML-Dateien zu verwenden.

Damit können Bio-Modelle leicht zwischen entsprechenden quantitativen Simulatoren und **BIOCHAM** ausgetauscht werden. Das folgende **Beispiel 5.6** illustriert die Verifikation von **Modell 1** bezüglich temporallogischer Eigenschaften mit Hilfe von **BIOCHAM**.

#### Beispiel 5.6 (Modell 1 in **BIOCHAM**)

In diesem Beispiel wird **BIOCHAM** benutzt, um temporale Eigenschaften von **Modell 1** zu verifizieren (siehe **Beispiel 5.5**). Dafür wurde die Online-Version von **BIOCHAM** 3.5 benutzt<sup>1</sup>. **Modell 1** ist bereits im Dateiformat von **BIOCHAM** verfügbar<sup>2</sup> und kann direkt über den Menüpunkt „Examples“ geladen werden. **Abbildung 5.6** zeigt eine grafische Darstellung der **BIOCHAM**-Regeln von **Modell 1**. Die Regeln repräsentieren dabei die Reaktionen von **Modell 1** (vgl. **Abbildung 1.3**). Reaktion 4 ist dabei über die beiden Regeln **rule\_4** und **rule\_5** repräsentiert, entsprechend der beiden Summanden des kinetischen Gesetzes (**Gleichung 1.14**). Die Substanzen haben Bezeichner, die sowohl deren Komponenten als auch ihre Phosphorylierung widerspiegeln. So ist etwa **Cdc2-Cyclin~{p1}** der Komplex aus **CDK1** (*Cyclin-dependent Kinase 1*) (hier entsprechend **Tyson 1991** als „Cdc2“ bezeichnet) und Cyclin, welches an Stelle 1 phosphoryliert ist. Dieser Komplex repräsentiert also den aktiven **MPF**.

In **BIOCHAM** müssen nicht notwendigerweise initiale Zustände (Startzustände der repräsentierten Kripke-Struktur) festgelegt werden. Temporallogische Aussagen in **BIOCHAM** sind deshalb stets bezüglich initialer Zustände quantifizierte **CTL**-Formeln. Dabei werden die beiden Quantoren *Ei* (es gibt einen initialen Zustand) und *Ai* (für alle initialen Zustände) verwendet. Eine (partielle) Definition eines initialen Zustands beschränkt diese Quantifizierung. Für die unten stehenden Anfragen wird der folgende initiale Zustand partiell festgelegt:

```
present(Cdc2,1).
present(Cdc2~{p1},0).
present(Cyclin,0).
present(Cdc2-Cyclin~{p1,p2},0).
present(Cdc2-Cyclin~{p1},0).
present(Cyclin~{p1},0).
```

In diesem initialen Zustand ist nur **CDK1** („Cdc2“) vorhanden (**present**). Alle anderen Konzentrationen sind auf 0 gesetzt.

Die Anfrage (**check\_ctl**) auf Erreichbarkeit von **MPF** (**Cdc2-Cyclin~{p1}**):

```
check_ctl(Ei(EF(Cdc2-Cyclin~{p1}))).
```

ergibt **true**, d.h. **MPF** kann von einem initialen Zustand aus produziert werden. Wird eine Begründung für diese Aussage verlangt (**why**), liefert **BIOCHAM** folgenden Pfad der Kripke-Struktur, für welchen **EF(Cdc2-Cyclin~{p1})** gilt:

```
1 Cdc2 is present
2 1 _=>Cyclin.
3 Cyclin is present
4 9 Cdc2=>Cdc2~{p1}.
5 Cdc2 is absent
```

<sup>1</sup><http://contraintes.inria.fr/biocham/online/>

<sup>2</sup>[http://contraintes.inria.fr/biocham/EXAMPLES/cell\\_cycle\\_Tyson\\_6Var/tyson91.bc](http://contraintes.inria.fr/biocham/EXAMPLES/cell_cycle_Tyson_6Var/tyson91.bc)

```

6 | Cdc2~{p1} is present
7 | 3 Cyclin+Cdc2~{p1}=>Cdc2-Cyclin~{p1,p2}.
8 | Cdc2~{p1} is absent
9 | Cyclin is absent
10| Cdc2-Cyclin~{p1,p2} is present
11| 4 Cdc2-Cyclin~{p1,p2}=>Cdc2-Cyclin~{p1}.
12| Cdc2-Cyclin~{p1,p2} is absent
13| Cdc2-Cyclin~{p1} is present

```

Dieser Pfad besteht wechselnd aus partiellen Zustandsbeschreibungen (eine oder mehrere Zeilen von atomaren Aussagen) und angewendeten Regeln (erkennbar an der vorangestellten Nummer der Regel und dem Punkt am Ende). Die Vorbedingungen einer Regel müssen an der Stelle der Regelanwendung gelten. Unterhalb einer Regelanwendung werden diejenigen atomaren Aussagen angeführt, die sich durch die Anwendung der Regel verändert haben. Z.B. ist die Vorbedingung für die Anwendung der Regel 9 (Zeile 4) in Zeile 1 gegeben. Durch die Anwendung von Regel 9 in Zeile 4 wird diese Vorbedingung negiert (Zeile 5). Am Ende dieses Pfades wird ein Zustand erreicht, in welchem **MPF** vorhanden ist (Zeile 13). Damit erfüllt der gesamte Pfad die gegebene **CTL**-Formel  $EF(Cdc2-Cyclin\sim\{p1\})$  für den initialen Zustand, in welchem mindestens **CDK1** („Cdc2“) vorhanden ist (Zeile 1).

Die Frage danach, ob ein stationärer Zustand bezüglich des Vorhandenseins von **MPF** ( $Cdc2-Cyclin\sim\{p1\}$ ) erreicht werden kann:

```
check_ctl(Ei(EF(AG(Cdc2-Cyclin~{p1}))))).
```

wird mit **false** beantwortet, während die abgeschwächte Frage (siehe [Beispiel 5.5](#))

```
check_ctl(Ei(EF(EG(Cdc2-Cyclin~{p1}))))).
```

**true** liefert. Dies ist nicht verwunderlich, da durch Anwendung einer Regel, die **MPF** zerstören kann (Regel 6 oder 7), jeweils eine Verzweigung im *Computation Tree* entsteht, wobei jeweils in einem der Folgezustände **MPF** vorhanden ist und im anderen nicht. Es gibt somit immer einen Pfad, in dem **MPF** stationär ist ( $EG(Cdc2-Cyclin\sim\{p1\})$ ). Jedoch ist nicht auf allen Pfaden **MPF** stets vorhanden,  $AG(Cdc2-Cyclin\sim\{p1\})$  gilt also nicht.

Die Frage nach einem zyklischen Verhalten von **MPF** ( $Cdc2-Cyclin\sim\{p1\}$ )

```
check_ctl(Ei(EG(Cdc2-Cyclin~{p1})->EF(!(Cdc2-Cyclin~{p1}))&
  (!(Cdc2-Cyclin~{p1})->EF(Cdc2-Cyclin~{p1}))))).
```

wird bejaht (**true**). Dabei steht **!** für die Negation, **&** für die Konjunktion und **->** für die Implikation. ■

### Fazit

Temporale Logiken sind in der Lage, Aussagen über zeitliche Zustandsfolgen (Pfade) zu formulieren. Dabei kann auch über alternative Pfade quantifiziert werden. Durch etablierte Verfahren des *Model Checkings* können so formulierte temporale Eigenschaften dynamischer Systeme automatisch verifiziert werden.

Leider gibt es bei der Modellierung durch Kripke-Strukturen das oben beschriebene „state explosion problem“ ([Abschnitt 5.3.1](#)) und den daraus resultierenden Berechnungsaufwand

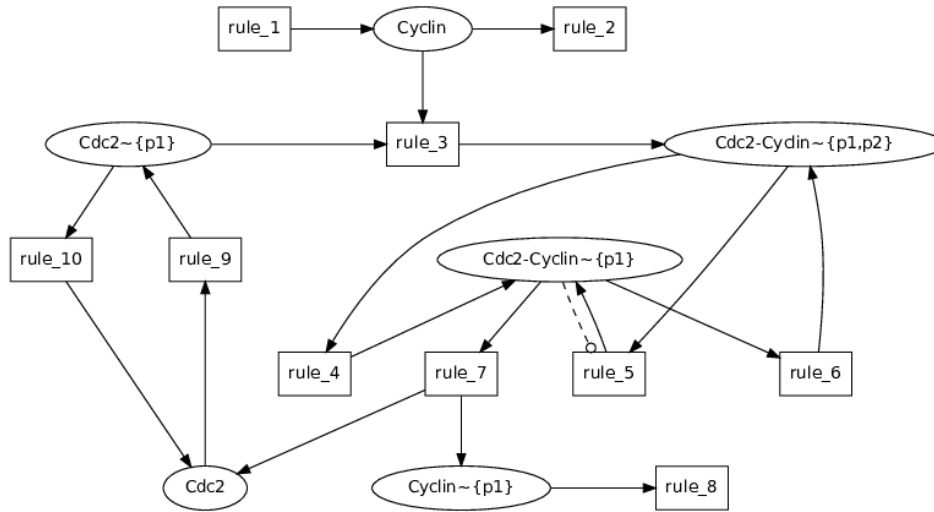


Abbildung 5.6.: Grafische Darstellung der [BIOCHAM](#)-Regeln für Modell 1 erzeugt mit der Online-Version von [BIOCHAM](#) 3.5. Dabei repräsentieren `rule_1` bis `rule_3` die Reaktionen 1 bis 3 aus [Abbildung 1.3](#), die `rule_4` und `rule_5` repräsentieren gemeinsam die Reaktion 4 und `rule_6` bis `rule_10` entsprechend die Reaktionen 5 bis 9. Es wurde eine zusätzliche Kante von `rule_5` auf `Cdc2-Cyclin~{p1}` zugefügt, die in der ursprünglichen Darstellung nicht vorhanden ist: Die gestrichelte Kante repräsentiert einen katalytischen Einfluss. Für Details siehe [Beispiel 5.6](#)

des *Model Checkings*. Darüber hinaus können mit temporalen Logiken nur sehr einfache Eigenschaften des Verhaltens dynamischer Systeme beschrieben werden. So sind etwa Bifurkationen oder asymptotische Verhaltensweisen nicht allein mit temporallogischen Mitteln charakterisierbar. Erfolgversprechend ist dagegen der Einsatz von temporalen Logiken und *Model Checking* als Teil integrierter Werkzeuge zur Modellierung und Simulation biologischer Systeme, wie etwa [BIOCHAM](#).

Eine detaillierte Charakterisierung der Dynamik eines Bio-Modells, wie sie durch die Theorie dynamischer Systeme ([Abschnitt 5.1](#)) ermöglicht wird, ist mit temporalen Logiken nicht möglich.

## Zusammenfassung

In diesem Kapitel wurden Ansätze für die qualitative Beschreibung der Dynamik von Bio-Modellen untersucht. Dabei zeigten sich verschiedene Stärken und Schwächen der einzelnen Ansätze. Sowohl in der qualitativen Modellierung als auch in Kripke-Strukturen wird der Zustand eines dynamischen Systems qualitativ abstrahiert. In qualitativen Simulationen werden mögliche Verhaltensweisen eines dynamischen Systems als Folgen qualitativer Zustände generiert ([Abschnitt 5.2](#)). Mit Hilfe temporaler Logiken können Aussagen über Zustandsfolgen formuliert und mit Techniken des *Model Checkings* überprüft werden ([Abschnitt 5.3](#)). Einen anderen Ansatz verfolgt die Theorie dynamischer Systeme ([Abschnitt 5.1](#)). Hier wird das Verhalten dynamischer Systeme durch wesentliche Merkmale von Phasenportraits charakterisiert. Außerdem werden qualitative Änderungen von Phasenportraits und ihre Abhängigkeit von Parametern untersucht. Ein Vorteil der Theorie dynamischer Systeme liegt sicher auch darin, dass Simulation und Analyse direkt auf das kontinuierliche Modell angewendet werden. Eine Transformation in eine QDE oder eine Kripke-Struktur ist dafür nicht notwendig. Die [Tabelle 5.1](#) fasst die Besonderheiten der vorgestellten Ansätze mit Blick auf die Beschreibung von Verhalten zusammen.

Dynamik-Merkmal	Theorie dynamischer Systeme	qualitative Modellierung und Simulation	temporale Logiken und <i>Model Checking</i>
Struktur	Differenzialgleichungen	QDEs (oder Ähnliches)	Kripke-Struktur
Zustand	reellwertig	diskret	diskret
Zeit	explizit	implizit	implizit
Berechnung Folgezustand	deterministisch	nichtdeterministisch	nichtdeterministisch
Verhalten	Trajektorie	Zustandsfolge	Zustandsfolge
alternative Verhalten	Phasenportrait	<i>Envisionment</i>	<i>Computation Tree</i>
Verhaltenscharakterisierung	Trajektorienart, Stabilität	—	temporale Logik (CTL*-Formeln)
Diversifikation	Bifurkation	Modellfragmente	—
Parametereinfluss	Bifurkationsdiagramm	Vorbedingungen für Modellfragmente	—

Tabelle 5.1.: Repräsentation unterschiedlicher Merkmale der Dynamik von Modellen in der Theorie dynamischer Systeme, in der qualitativen Modellierung und Simulation sowie in temporalen Logiken und dem *Model Checking*.

Die vorgestellten Verfahren für die qualitative Beschreibung der Dynamik von Modellen können ihre Vorzüge bei unterschiedlichen Aufgaben der Modellentwicklung (vgl. [Abschnitt 1.3.1](#) und [Abschnitt 1.3.2](#)) ausspielen. Die Techniken der Modellkomposition der

qualitativen Modellierung sind bei der Modellbildung und Modellmodifikation hilfreich. Die qualitative Simulation erlaubt schnelle Aussagen darüber, ob ein Modell prinzipiell in der Lage ist, die Dynamik des modellierten Systems abzubilden. Dies kann für die Modellbewertung und für die Modellevolution nützlich sein. *Model Checking* kann eingesetzt werden, um temporale Eigenschaften des Modellverhaltens zu verifizieren und somit die Modellsuche und die diagnosebasierte Modellrevision zu leiten. Schließlich kann die Theorie dynamischer Systeme für eine tiefgehende Analyse und Charakterisierung der Dynamik ausgewählter Modelle verwendet werden.

Die Konzepte und Verfahren sowohl der qualitativen Modellierung und Simulation als auch der temporalen Logiken und des *Model Checkings* sind hinreichend formalisiert und in Form von Algorithmen realisiert. Dagegen sind zwar die Konzepte der Theorie dynamischer Systeme mathematisch definiert und für grundlegende Analyseaufgaben existieren auch Algorithmen. Für eine formale Repräsentation der Dynamik von Bio-Modellen auf der Basis der Theorie dynamischer Systeme fehlen jedoch adäquate Ausdrucksmittel. Dies erschwert die Einbindung von Prinzipien der Theorie dynamischer Systeme in Verfahren für die computergestützte Modellierung und Simulation. Als ein wichtiger Schritt für eine solche formale Repräsentation wird mit [TEDDY](#) (*TErminology for the Description of DYnamics*, [COURTOT ET AL. 2011](#)) eine Ontologie entwickelt, die wesentliche Konzepte der Theorie dynamischer Systeme begrifflich formalisiert. Damit beschäftigt sich das folgende [Kapitel 6](#).





## 6. Terminologie dynamischen Verhaltens: TEDDY

„We can roughly classify all possible orbits in dynamical systems into fixed points, cycles, and ,all others.“

---

(KUZNETSOV 1998: *Elements of applied bifurcation theory*, S. 10)

Im vorhergehenden Kapitel 5 wurden Ansätze vorgestellt, mit denen sich das Verhalten von Bio-Modellen qualitativ beschreiben lässt: Die Theorie dynamischer Systeme beschreibt die Formen zeitlicher Entwicklungen sowie deren Abhängigkeiten von Parametern, die qualitative Modellierung erlaubt qualitative Aussagen über Zustände und Zustandsübergänge und mit Hilfe temporaler Logiken können Zustandsfolgen charakterisiert werden. Für die Formalisierung der Dynamik von Bio-Modellen und deren Diversifikationen (Facette B3, siehe Abschnitt 2.2.3) sind diese Ansätze mehr oder weniger hilfreich: Die qualitative Modellierung und die temporalen Logiken können Diversifikationen nicht adäquat repräsentieren, die Theorie dynamischer Systeme ist begrifflich nicht formalisiert. Im Rahmen dieser Arbeit wurde die Ontologie TEDDY (*TErminology for the Description of DYnamics*, COURTOT ET AL. 2011) entwickelt, um eine formale begriffliche Basis für die Theorie dynamischer Systeme bereitzustellen. Dies stellt einen wichtigen Grundbaustein für die Formalisierung der Dynamik von Bio-Modellen dar.

In Abschnitt 6.1 werden zunächst Aufbau und Inhalt von TEDDY präsentiert. Dabei wird gezeigt, wie die Grundkonzepte der Theorie dynamischer Systeme, insbesondere Typen von Trajektorien und Bifurkationen, konzeptuell erfasst werden. TEDDY wurde (und wird weiterhin) als frei verfügbare Ontologie als Community-Projekt entwickelt und in OWL (*Web Ontology Language*, HITZLER ET AL. 2012) spezifiziert. In Abschnitt 6.2 werden pragmatische Aspekte der Entwicklung von TEDDY beleuchtet. Zum Abschluss werden noch Hinweise zur Verwendung von TEDDY gegeben (Abschnitt 6.3).

Die bei der Beschreibung des Inhalts von TEDDY verwendeten deutschen Bezeichnungen für Konzepte der Theorie dynamischer Systeme folgen im Wesentlichen JETSCHKE 1989. Die TEDDY-Konzepte selbst tragen geeignete englische Bezeichnungen.

Die folgenden Ausführungen beziehen sich, falls nicht anderweitig angegeben, auf die aktuelle TEDDY-Release rel-2014-04-24. In Anhang F ist die gesamte Klassenhierarchie dieser TEDDY-Release dargestellt, die vollständige OWL-Codierung findet sich unter: <http://identifiers.org/combine.specifications/teddy.rel-2014-04-24.owl> .

### 6.1. Inhalt von TEDDY

In diesem Abschnitt wird die Ontologie TEDDY in der aktuellen Release rel-2014-04-24 vorgestellt. TEDDY bietet Konzepte, mit deren Hilfe die Dynamik von Bio-Modellen

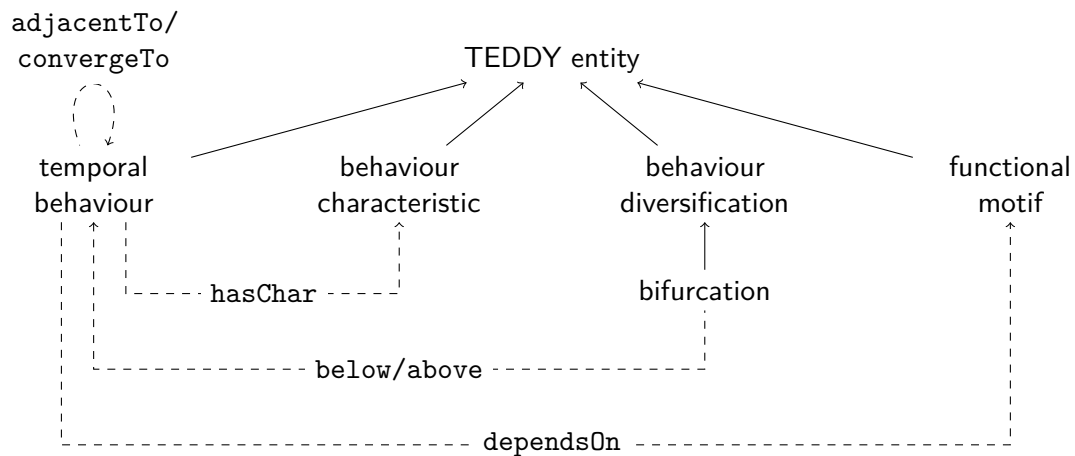


Abbildung 6.1.: Obere Ebenen von **TEDDY**: Unterkonzepte von 'TEDDY entity' [[TEDDY\\_0000000](#)] und wichtige Relationen. Durchgezogene Kanten repräsentieren Subsumtionsbeziehungen (`rdfs:SubClassOf`), entsprechend beschriftete gestrichelte Kanten andere Relationen vom Typ `owl:ObjectProperty` (`hasChar` abkürzend für `hasCharacteristic`). Der Zweig für veraltete Konzepte ('\_obsolete' [[TEDDY\\_obsolete](#)]) ist nicht dargestellt. Für Details siehe Text.

beschrieben werden kann. In [Abbildung 6.1](#) sind die oberen zwei Ebenen von **TEDDY** sowie die wichtigsten Relationen dargestellt. Als gemeinsames Oberkonzept für die verschiedenen dynamischen Aspekte dient das sehr allgemeine Konzept 'TEDDY entity' [[TEDDY\\_0000000](#)]. Konzepte in **TEDDY** haben eine eindeutige siebenstellige ID, welche in dieser Arbeit in eckigen Klammern nach der Bezeichnung der Konzepte angegeben wird (siehe auch [Abschnitt 6.2.2](#)). Alle angeführten **TEDDY**-Konzepte sind außerdem in [Anhang B](#) aufgelistet.

In den folgenden Abschnitten werden die Konzepte zur Charakterisierung zeitlichen Verhaltens ([Abschnitt 6.1.1](#)), für die Beschreibung der Diversifikation von Verhalten ([Abschnitt 6.1.2](#)) und für funktionale Struktur-Muster in dynamischen Systemen ([Abschnitt 6.1.3](#)) im Detail eingeführt. Pragmatische Aspekte, wie die Geschichte, der Entwicklungsprozess und technische Bemerkungen zur Implementierung von **TEDDY** werden im darauffolgenden [Abschnitt 6.2](#) diskutiert.

### 6.1.1. Zeitliches Verhalten

Offensichtliche Konzepte, die zur qualitativen Beschreibung von Verhalten benutzt werden, sind solche, die wiederkehrende Muster in zeitlichen Verläufen (Trajektorien) *phänomenologisch* charakterisieren, wie etwa „Oszillation“, „stationärer Zustand“ und „monoton steigend/fallend“ (vgl. [Abschnitt 5.1.2](#)). Die zentrale Charakterisierung von Trajektorien in der Theorie dynamischer Systeme abstrahiert jedoch vom eigentlichen zeitlichen Verlauf und betrachtet das Langzeitverhalten. Trajektorien mit gleichem Langzeitverhalten werden als „topologische Typen“ ([JETSCHKE 1989](#), S. 38ff) bezeichnet. Beispiele für topologische Typen sind „Fixpunkt“, „Zyklus“, „Spirale“. Die Charakterisierung des Langzeitverhaltens schließt eine Betrachtung der Stabilität gegenüber kleinen Störungen ein, z.B. durch die Begriffe „asymptotisch stabil“ und „instabil“. Außerdem wird das Verhalten von Trajekto-

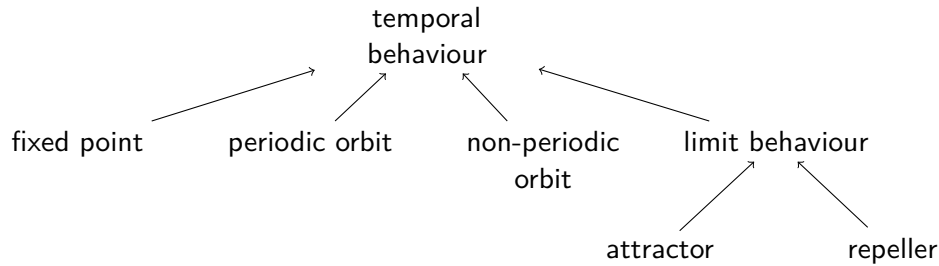


Abbildung 6.2.: TEDDY-Zweig 'temporal behaviour' [TEDDY\_0000083]. Kanten repräsentieren Subsumtionsbeziehungen (`rdfs:SubClassOf`). Für Details siehe Text.

rien untersucht, die in der Nähe von Fixpunkten oder von Zyklen starten. Dabei können die Fixpunkte bzw. Zyklen z.B. „Grenzmenge“, „Attraktor“ und „Repeller“ sein.

### TEDDY-Zweig 'temporal behaviour'

Der wichtigste TEDDY-Zweig 'temporal behaviour' [TEDDY\_0000083] klassifiziert die topologischen Typen von Trajektorien, d.h. das Langzeitverhalten, in Fixpunkte ('fixed point' [TEDDY\_0000086]), geschlossene Trajektorien ('periodic orbit' [TEDDY\_0000050]) sowie Trajektorien, die unter keine der anderen Klassen fallen ('non-periodic orbit' [TEDDY\_0000143]). Daneben gibt es noch ein weiteres Konzept, die Grenzmengen ('limit behaviour' [TEDDY\_0000085]). Zu den nicht-periodischen Trajektorien zählen neben den asymptotisch auf Grenzmengen zulaufenden Trajektorien ('asymptotic behaviour' [TEDDY\_0000146]) auch chaotische Attraktoren ('strange attractor' [TEDDY\_0000054]) und Verbindungen zwischen Sattelpunkten ('saddle connection' [TEDDY\_0000099]; vgl. STROGATZ 2001). Die Einteilung in Fixpunkte, geschlossene und nicht-periodische Trajektorien folgt einer Einteilung möglicher Periodenlängen (Dauer bis zur Wiederkehr eines Systemzustandes): Während geschlossene Trajektorien eine endliche Periodenlänge haben, ist die Periodenlänge von Fixpunkten null und die von nicht-periodischen Trajektorien unendlich. Die oberen Ebenen des TEDDY-Zweigs 'temporal behaviour' [TEDDY\_0000083] sind in Abbildung 6.2 dargestellt.

Das Konzept der Grenzmenge ('limit behaviour' [TEDDY\_0000085]; vgl. JETSCHKE 1989, S. 54) ist ein Oberkonzept für Attraktoren ('attractor' [TEDDY\_0000094]) und Repeller ('repeller' [TEDDY\_0000102]). Darunter fallen bestimmte Trajektorien, die bereits in den angeführten Unterzweigen von 'temporal behaviour' [TEDDY\_0000083] vorkommen, und zwar stabile bzw. instabile Fixpunkte und Grenzzyklen ('limit cycle' [TEDDY\_0000051]) sowie chaotische Attraktoren ('strange attractor' [TEDDY\_0000054]).

In TEDDY stehen zwei Relationen zur Verfügung, um Beziehungen zwischen Trajektorien auszudrücken: Zwei Trajektorien können nahe beieinander liegen (`adjacentTo`) oder eine Trajektorie kann gegen eine andere Trajektorie konvergieren (`convergeTo`). Beide Relationen sind irreflexiv, sie beschreiben also *echte* Nachbarschaft bzw. Konvergenz. Die Relation `adjacentTo` ist symmetrisch. Die Relationen werden verwendet, um bestimmte Trajektorien (partiell) zu definieren. So ist z.B. ein Grenzzyklus ('limit cycle' [TEDDY\_0000051]) eine isolierte geschlossene Trajektorie (STROGATZ 2001, S. 196), also ein 'periodic orbit' [TEDDY\_0000050], welcher nicht von anderen geschlossenen Trajektorien umgeben ist (alle folgenden TEDDY-Axiome sind in Manchester Syntax, HORRIDGE ET AL. 2012 gegeben):

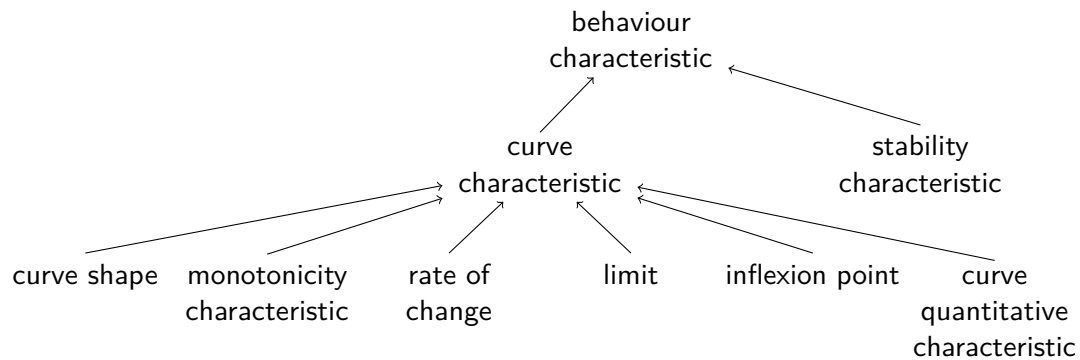


Abbildung 6.3.: TEDDY-Zweig 'behaviour characteristic' [TEDDY\_0000002]. Kanten repräsentieren Subsumtionsbeziehungen (`rdfs:SubClassOf`). Für Details siehe Text.

```

Class: 'limit cycle'
SubClassOf: 'periodic orbit',
            and not (adjacentTo some 'periodic orbit')

```

Und eine asymptotische Trajektorie ('asymptotic behaviour' [TEDDY\_0000146]) konvergiert gegen ein Grenzverhalten ('limit behaviour' [TEDDY\_0000085]):

```

Class: 'asymptotic behaviour'
SubClassOf: 'non-periodic orbit'
            and (convergeTo some 'limit behaviour')

```

### TEDDY-Zweig 'behaviour characteristic'

Trajektorien ('temporal behaviour' [TEDDY\_0000083]) können in TEDDY mit Hilfe der Relation `hasCharacteristic` durch Konzepte aus dem Zweig 'behaviour characteristic' [TEDDY\_0000002] näher charakterisiert werden. In Abbildung 6.3 sind die Unterkonzepte von 'behaviour characteristic' [TEDDY\_0000002] dargestellt, die im Folgenden besprochen werden.

Für die phänomenologische Charakterisierung zeitlichen Verhaltens (siehe Abschnitt 5.1.2) dienen die Konzepte unter 'curve characteristic' [TEDDY\_0000023]. Damit kann unter anderem die Form ('curve shape' [TEDDY\_0000015]), die Richtung ('monotonicity characteristic' [TEDDY\_0000144]) und die Stärke des Wachstums ('rate of change' [TEDDY\_0000012]) der grafischen Darstellung von Zeitverläufen beschrieben werden. Außerdem kann spezifiziert werden, ob eine Trajektorie einem Grenzwert zustrebt ('limit' [TEDDY\_0000025]) oder einen Wendepunkt besitzt ('inflexion point' [TEDDY\_0000130]). Weiterhin können konkrete Trajektorien durch Konzepte unter 'curve quantitative characteristic' [TEDDY\_0000178] quantitativ charakterisiert werden. So kann etwa die Lage eines stationären Zustands durch 'magnitude' [TEDDY\_0000116] gegeben werden und die Periode und Amplitude einer Oszillation durch 'period' [TEDDY\_0000067] und 'amplitude' [TEDDY\_0000131]. Diese Konzepte verlangen die Angabe eines entsprechenden numerischen Wertes (vom Typ `xsd:double`) mit Hilfe von `hasValue`, einer `DataProperty`.

Eine wichtige Charakterisierung von Fixpunkten und geschlossenen Trajektorien ist deren Stabilität gegenüber kleineren Störungen (siehe [Abschnitt 5.1](#)). Für diese Charakterisierung dienen die Konzepte unter 'stability characteristic' [[TEDDY\\_0000148](#)]. Diese Konzepte sind allgemein für invariante Mengen definiert. Die Stabilität von Fixpunkten und geschlossenen Trajektorien kann auf verschiedene Weisen in einer Ontologie repräsentiert werden. In [Abschnitt 6.2.3](#) wird ausführlich dargelegt, warum in [TEDDY](#) diese Variante gewählt wurde.

Die Konzepte für die Stabilität von invarianten Mengen werden in [TEDDY](#) benutzt, um bestimmte Fixpunkte und geschlossene Trajektorien zu definieren. So ist zum Beispiel ein stabiler Grenzzyklus ('stable limit cycle' [[TEDDY\\_0000114](#)]) partiell definiert als ein:

```
Class: 'stable limit cycle'
  SubClassOf: 'limit cycle',
    and hasCharacteristic some 'asymptotic stability'
```

Mit Hilfe der Konzepte für die Stabilitätseigenschaften ('stability characteristic' [[TEDDY\\_0000148](#)]) werden die Grenzverhalten ('limit behaviour' [[TEDDY\\_0000085](#)]) definiert, die eine sekundäre Klassifikation von Trajektorien erlauben (vgl. [Abschnitt 6.2.3](#)).

### 6.1.2. Diversifikation von Verhalten

Der [TEDDY](#)-Zweig 'temporal behaviour' [[TEDDY\\_0000083](#)] (siehe [Abschnitt 6.1.1](#)) beschreibt konkrete zeitliche Verhaltensweisen eines dynamischen Systems. Es gibt aber auch System-Eigenschaften, die das Verhalten eines dynamischen Systems gegenüber Störungen charakterisieren, d.h. die Diversifikation seiner Dynamik (vgl. [Facette B3](#)), in [TEDDY](#) repräsentiert als Unterkonzepte von 'behaviour diversification' [[TEDDY\\_0000132](#)]. Die Störungen können sich dabei auf den Zustand des Systems beziehen oder auf seine Parameter. Störungen der ersten Art entstehen im biologischen Zielsystem z.B. durch die Zugabe einer Substanz, wodurch sich deren Konzentration ändert. In [TEDDY](#) werden die verschiedenen Weisen, auf solche Störungen zu reagieren durch Konzepte unter 'perturbation response' [[TEDDY\\_0000108](#)] klassifiziert. Ein Beispiel für eine Störung der zweiten Art ist eine Temperaturänderung im biologischen Zielsystem, durch die sich die kinetischen Parameter ändern. Diese Art qualitativer Änderungen der Systemdynamik durch Parameteränderungen wird im [TEDDY](#)-Zweig 'bifurcation' [[TEDDY\\_0000053](#)] charakterisiert. Als dritte Art der Diversifikation gibt es das sogenannte „*Bursting*“ ('bursting' [[TEDDY\\_0000055](#)]), welches insbesondere bei der Modellierung von Neuronen eine Rolle spielt. Als *Burst* wird eine schnelle Folge von *Spikes* (abrupter kurzzeitiger Anstieg des Membranpotentials eines Neuron) bezeichnet. Beim *Bursting* wechseln sich Ruhephasen mit *Bursts* ab. Dieser Vorgang wird durch zwei interagierende Teilsysteme modelliert: Das „schnelle“ Teilsystem bringt durch einen Grenzzyklus die *Bursts* hervor, während das „langsame“ Teilsystem diesen Vorgang moduliert, indem es einen kritischen Parameter des schnellen Teilsystems ändert. Dabei spielen beide Arten von Störungen eine Rolle: Bifurkationen des schnellen Teilsystems erwirken den Wechsel zwischen Ruhephase und *Burst*, eine Bistabilität (siehe unten) des langsamen Teilsystems kann diese Bifurkationen auslösen (für Details siehe etwa [IZHIKEVICH 2007](#)). Es gibt eine Vielzahl verschiedener Arten des *Burstings*. Entsprechende Unterkonzepte von 'bursting' [[TEDDY\\_0000055](#)] in [TEDDY](#) fehlen jedoch zur Zeit. Die Klassifikation in [IZHIKEVICH 2007](#) (S. 347ff) bietet einen guten Einstiegspunkt für einen

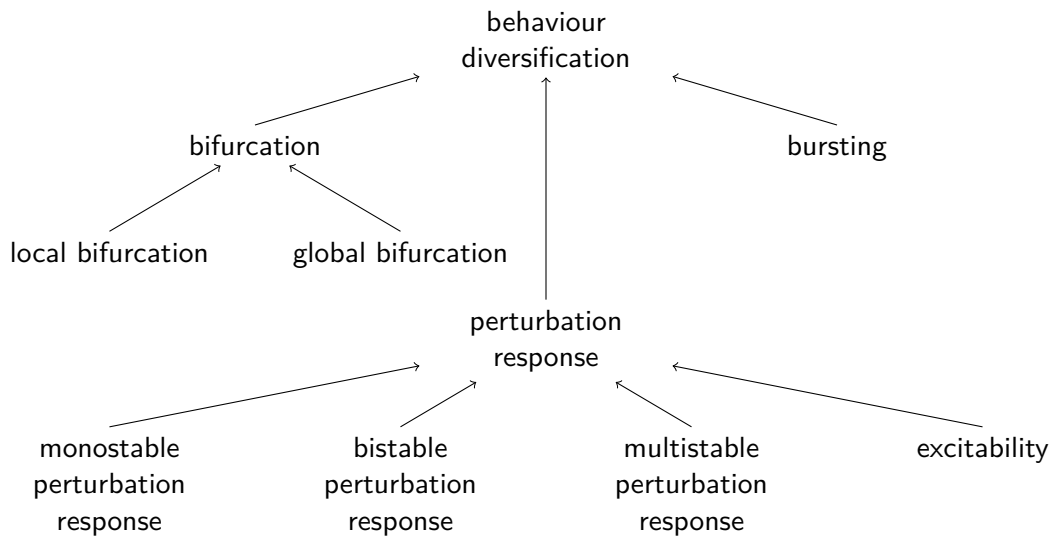


Abbildung 6.4.: TEDDY-Zweig 'behaviour diversification' [TEDDY\_0000132]. Kanten repräsentieren Subsumtionsbeziehungen (`rdfs:SubClassOf`). Für Details siehe Text.

zukünftigen Ausbau dieses TEDDY-Zweiges. In [Abbildung 6.4](#) ist die obere Ebene des TEDDY-Zweiges 'behaviour diversification' [TEDDY\_0000132] dargestellt, welcher Konzepte für die Charakterisierung der Diversifikation eines dynamischen Systems beinhaltet.

### TEDDY-Zweig 'perturbation response'

Ein dynamisches System kann auf verschiedene Arten auf Störungen seines Zustands reagieren. Eine wichtige Rolle spielen dabei stabile Trajektorien, wie stabile Fixpunkte und stabile geschlossene Trajektorien (vgl. [Abschnitt 6.1.1](#)). Bei hinreichend kleinen Störungen eines stabilen Zustands kehrt das System in diesen stabilen Zustand zurück. Ein dynamisches System besitzt eine charakteristische Anzahl stabiler Trajektorien. Der einfachste Fall eines Systems mit nur einer stabilen Trajektorie wird in TEDDY durch das Konzept 'monostable perturbation response' [TEDDY\_0000109] erfasst. Interessanter sind Systeme, die zwei oder mehrere stabile Trajektorien besitzen. Überschreiten die Störungen eine bestimmte Schwelle, wird eine andere stabile Trajektorie erreicht. Dadurch kann sich das Verhalten des Systems radikal ändern, d.h. es tritt eine Diversifikation des Verhaltens auf. So kann z.B. ein Übergang von einem stabilen Fixpunkt mit einem stationären Zustand zu einem stabilen Grenzzyklus mit oszillierendem Verhalten erfolgen. Systeme mit zwei stabilen Trajektorien zeigen eine 'bistable perturbation response' [TEDDY\_0000110], Systeme mit mehr als zwei stabilen Trajektorien eine 'multistable perturbation response' [TEDDY\_0000111].

In TEDDY wird die Stabilität gegenüber kleinen Zustandsstörungen auf zwei unterschiedliche Weisen konzeptualisiert: Zum einen gibt es die Stabilitätseigenschaften von invarianten Mengen ('stability characteristic' [TEDDY\_0000148], vgl. [Abschnitt 6.1.1](#)). Zum anderen werden, wie hier beschrieben, unter 'perturbation response' [TEDDY\_0000108] die verschiedenen Weisen charakterisiert, wie dynamische Systeme auf Störungen reagieren können. Letzteres charakterisiert also ein dynamisches System als Ganzes.

### TEDDY-Zweig 'bifurcation'

Bifurkationen des dynamischen Systems können mit Konzepten unter 'bifurcation' [TEDDY\_0000053] spezifiziert werden. Es gibt sowohl Konzepte für lokale Bifurkationen von Fixpunkten ('local bifurcation' [TEDDY\_0000069]), welche durch Linearisierung in der Umgebung des Fixpunktes analysiert werden können (vgl. KUZNETSOV 1998, S. 58), als auch für globale Bifurkationen ('global bifurcation' [TEDDY\_0000147]), für deren Analyse es nicht ausreicht, lediglich die Umgebung des Fixpunktes zu betrachten. In globale Bifurkationen sind geschlossene Trajektorien oder mehrere Fixpunkte involviert.

Bei einer Bifurkation bezüglich eines Parameters gibt es eine topologische Veränderung im Phasenportrait: Phasenportraits unterhalb des Bifurkationspunktes unterscheiden sich qualitativ von Phasenportraits oberhalb des Bifurkationspunktes. Die verschiedenen Arten von Bifurkationen sind durch charakterisierende Trajektorien der Phasenportraits unterhalb bzw. oberhalb des Bifurkationspunktes gekennzeichnet. In TEDDY wird dies durch die entsprechenden Relationen `below` und `above` zwischen Bifurkationen ('bifurcation' [TEDDY\_0000053]) und diese charakterisierende Trajektorien ('temporal behaviour' [TEDDY\_0000083]) repräsentiert. Für den kritischen Parameterwert besitzt das dynamische System ebenfalls ein Phasenportrait. Dessen Charakterisierung ist für die Definition einer Bifurkationsart jedoch unbedeutend (siehe auch Beispiel 6.1), weshalb auf eine diesbezügliche Darstellung in TEDDY verzichtet wurde.

#### Beispiel 6.1 (Definition von Hopf-Bifurkationen in TEDDY)

Eine Hopf-Bifurkation ('Hopf bifurcation' [TEDDY\_0000072]) ist definiert als lokale Bifurkation ('local bifurcation' [TEDDY\_0000069]), deren Phasenportrait unterhalb des Bifurkationspunktes einen stabilen Strudel ('stable spiral' [TEDDY\_0000126]) und oberhalb des Bifurkationspunktes einen instabilen Strudel ('unstable spiral' [TEDDY\_0000127]) aufweist:

```
Class: 'Hopf bifurcation'
  SubClassOf: 'local bifurcation',
              and (below some 'stable spiral')
              and (above some 'unstable spiral')
```

Es gibt drei Arten von Hopf-Bifurkationen, die sich dadurch unterscheiden, ob und für welche Parameterwerte im Phasenportrait ein Grenzzyklus auftritt: (1) Bei der subkritischen Hopf-Bifurkation ('subcritical Hopf bifurcation' [TEDDY\_0000073]) existiert ein instabiler Grenzzyklus ('unstable limit cycle' [TEDDY\_0000128]) unterhalb des Bifurkationspunktes:

```
Class: 'subcritical Hopf bifurcation'
  SubclassOf: 'Hopf bifurcation'
              and (below some 'unstable limit cycle')
```

(2) Dagegen existiert bei einer superkritischen Hopf-Bifurkation 'supercritical Hopf bifurcation' [TEDDY\_0000074] ein stabiler Grenzzyklus ('stable limit cycle' [TEDDY\_0000114]) oberhalb des Bifurkationspunktes:

```
Class: 'supercritical Hopf bifurcation'
  SubclassOf: 'Hopf bifurcation'
              and (above some 'stable limit cycle')
```



(3) Bei einer degenerierten Hopf-Bifurkation ('degenerate Hopf bifurcation' [TEDDY\_0000120]) existiert weder unterhalb noch oberhalb des Bifurkationspunktes ein Grenzzyklus ('limit cycle' [TEDDY\_0000051]):

```
Class: 'degenerate Hopf bifurcation'
SubclassOf: 'Hopf bifurcation'
and not (above some 'limit cycle')
and not (below some 'limit cycle')
```

Eine degenerierte Hopf-Bifurkation besitzt keine charakterisierenden Trajektorien unterhalb bzw. oberhalb des Bifurkationspunktes, wie die beiden anderen Hopf-Bifurkation. Aus diesem Grund beinhaltet die Definition der degenerierten Hopf-Bifurkation in TEDDY Negationen. Zur Vermeidung dieser Negationen kann das Phasenportrait für den kritischen Parameterwert betrachtet werden: Bei einer degenerierten Hopf-Bifurkation existiert am Bifurkationspunkt ein Wirbel ('center' [TEDDY\_0000088]), welcher von unendlich vielen nicht-isolierten geschlossenen Trajektorien ('non-isolated cycle' [TEDDY\_0000106]) mit steigender Periode umgeben ist. Da der Bifurkationspunkt numerisch meist nur näherungsweise bestimmt werden kann, ist eine Analyse des Phasenportraits am Bifurkationspunkt nur eingeschränkt möglich. Deshalb und weil einzig die degenerierte Hopf-Bifurkation eine Betrachtung des Phasenportraits für den kritischen Parameterwert notwendig macht, wurde in TEDDY auf die Einführung einer entsprechenden zusätzlichen Relation (etwa `at`) zwischen Bifurkationen und diese charakterisierende Trajektorien des Phasenportraits für den kritischen Parameterwert verzichtet. ■

Für die Relationen zwischen Bifurkationen und diese charakterisierende Phasenportrait-Merkmale, die unterhalb bzw. oberhalb eines Bifurkationspunktes bestehen, wurden in TEDDY die oben verwendeten Bezeichnungen gewählt: `below` und `above`. Die Wahl von passenden – d.h. gebräuchlicher und sprachlich adäquater – Bezeichnern für diese Relationen ist schwierig, wie die folgenden Betrachtungen zeigen. In der Literatur finden sich zahlreiche bildhafte Bezeichnungen, die eine Bifurkation als einen dynamischen Prozess auffassen, wie etwa „before“ und „after“. In diesem vorgestellten Prozess der Bifurkation werden dann Merkmale des Phasenraums – wie es etwa bei KUZNETSOV 1998 heißt – „zerstört“ („destroyed“) und „erzeugt“ („created“) oder noch malerischer:

„The birth, or the death, of a periodic orbit through a change in the stability of an equilibrium point is known as the Poincaré-Andronov-Hopf bifurcation.“  
(HALE ET AL. 1991, S. 208)

KUZNETSOV 1998 (S. 89) kritisiert die Verwendung von zeitlichen Bezeichnungen wie „before“ und „after“, da sie von der Richtung der Parametervariation abhängen. Um diese Abhängigkeit zu vermeiden, kann von „unterhalb“ („below“) und „oberhalb“ („above“) des Bifurkationspunktes gesprochen werden, wie z.B. STROGATZ 2001 (S. 59) dies tut. Diese Charakterisierung bezieht sich direkt auf den Parameterwert  $\mu$  im Vergleich zum Bifurkationspunkt  $\mu_{critical}$ : Ein Phasenportrait ist *unterhalb* der Bifurkation, falls  $\mu < \mu_{critical}$  und *oberhalb* einer Bifurkation, falls  $\mu > \mu_{critical}$ . Damit werden die Relationen unabhängig vom zeitlichen Prozess des Stattfindens einer Bifurkation und damit von der Richtung der Parameteränderung.

Eine alternative Bezeichnung für unterhalb bzw. oberhalb des kritischen Bifurkationspunktes (oft auch „kritischer“ Parameterwert genannt) ist subkritisch bzw. superkritisch.

Diese beiden Bezeichnungen dienen jedoch meist zur Unterscheidung von verschiedenen Ausprägungen einer Bifurkation, wie im Beispiel der Hopf-Bifurkation: der Grenzzyklus existiert oberhalb des Bifurkationspunktes bei der superkritischen Hopf-Bifurkation und unterhalb des Bifurkationspunktes bei der subkritischen Hopf-Bifurkation (vgl. [STROGATZ 2001](#), S. 59 und [KUZNETSOV 1998](#), S. 89). Auch diese Bezeichnungen werden von [KUZNETSOV 1998](#) (S. 127) kritisiert und stattdessen die Bezeichnungen „soft“ oder „noncatastrophic“ für die superkritische Hopf-Bifurkation und „sharp“ bzw. „catastrophic“ für die subkritische Hopf-Bifurkation vorgeschlagen, um die Folgen des Stabilitätsverlustes des Fixpunktes deutlich zu machen: bei der nicht-katastrophalen Hopf-Bifurkation sind die vom Fixpunkt wegstrebbenden Trajektorien durch den Grenzzyklus beschränkt, während im katastrophalen Fall sich die Trajektorien unendlich weit vom Fixpunkt entfernen ([KUZNETSOV 1998](#), S. 89).

### 6.1.3. Funktionale Struktur-Muster

Neben Konzepten für das zeitliche Verhalten eines dynamischen Systems und für die Diversifikation des zeitlichen Verhaltens gibt es in [TEDDY](#) auch Konzepte, die wiederkehrende funktionale Teilsysteme beschreiben, wie etwa die wohlbekannte negative Rückkopplung (siehe z.B. [WOLKENHAUER ET AL. 2005](#)). Solche „network motifs“ ([ALON 2007a](#)) sind auf den ersten Blick strukturelle Muster in dynamischen Systemen. Es gibt jedoch verschiedene Gründe, diese vermeintlich strukturellen Muster in [TEDDY](#) aufzunehmen: (1) Eine ontologische Analyse zeigt, dass diese Muster nicht ausschließlich strukturell sind. Mit „negativer Rückkopplung“ wird nicht die entsprechende Struktur im dynamischen System benannt, sondern vielmehr der Prozess des Rückwirkens, der durch diese Struktur vermittelt wird. (2) Die Definition dieser Muster kann nicht ausschließlich strukturell erfolgen. Zum Beispiel kann eine Rückkopplung abhängig von bestimmten Parametern entweder positiv oder negativ sein. Zumindest sind also funktionale Kenntnisse bei der Beschreibung der Muster involviert, weshalb die Muster auch als „funktionale“ Muster bezeichnet werden. (3) Das vielleicht gewichtigste Argument ist pragmatischer Natur. [TEDDY](#) soll der Beschreibung der Dynamik von Bio-Modellen dienen. Die funktionalen Muster werden oft genau dafür verwendet: mit ihrer Hilfe wird das Verhalten von Modellen charakterisiert. So werden etwa in [ALON 2007a](#), [ALON 2007b](#) und [WOLKENHAUER ET AL. 2005](#) die funktionalen Muster ganz explizit besprochen. Aber auch für die Charakterisierung der Dynamik konkreter Modelle finden die funktionalen Muster Anwendung. Im folgenden Zitat etwa ist die negative Rückkopplung ganz deutlich eine dynamische Charakteristik:

„[...] the cell cycle is driven by continuous biochemical oscillations and that the triggering by cdc2 kinase of cyclin degradation creates a negative feedback loop on which a minimum cell-cycle oscillator could be based [...].“ ([GOLDBETER 1991](#), S. 9110)

Unter 'functional motif' [[TEDDY\\_0000003](#)] finden sich in [TEDDY](#) entsprechende Konzepte für funktionale Strukturmuster, wie etwa Rückkopplungen ('feedback loop' [[TEDDY\\_0000033](#)]) und Vorwärtskopplungen ('feed-forward loop' [[TEDDY\\_0000036](#)]). Die Klassifikation der funktionalen Struktur-Muster in [TEDDY](#) geht im Wesentlichen auf die Arbeiten von ALON zurück ([ALON 2007b](#) und [ALON 2007a](#)).

Es gibt funktionale Muster, deren Existenz notwendige Voraussetzung für bestimmtes zeitliches Verhalten ist. So erfordert ungedämpfte Oszillation basierend auf einem stabilen

Grenzzzyklus ('stable limit cycle' [TEDDY\_0000114]) eine negative Rückkopplung ('negative feedback loop' [TEDDY\_0000034]). In TEDDY kann dies mit Hilfe der Relation `dependsOn` ausgedrückt werden:

```
Class: 'stable limit cycle'  
SubClassOf: dependsOn some 'negative feedback loop'
```

## 6.2. Entwicklung von TEDDY

Nachdem im vorhergehenden [Abschnitt 6.1](#) der Inhalt von TEDDY beschrieben wurde, soll nun auf einige pragmatische Aspekte der Entwicklung und Verwendung der Ontologie eingegangen werden. Zunächst wird die Geschichte von TEDDY beleuchtet ([Abschnitt 6.2.1](#)) sowie einige technische Aspekte der Entwicklung von TEDDY erläutert ([Abschnitt 6.2.2](#)). Im Entwicklungsprozess von TEDDY galt es immer wieder, Designentscheidungen zu treffen. Dabei musste zwischen ontologischen, repräsentationalen und pragmatischen Überlegungen abgewogen werden. Im Rahmen dieser Arbeit ist es weder möglich noch beabsichtigt, all diese Entscheidungen zu diskutieren. Stattdessen werden anhand der Repräsentation von Stabilitätseigenschaften einige wichtige und typische Designentscheidungen und dabei auftretende Probleme bei der Entwicklung von TEDDY beschrieben ([Abschnitt 6.2.3](#)).

### 6.2.1. Entstehungsgeschichte von TEDDY

Die Idee zur Entwicklung einer Ontologie für die Beschreibung des Verhaltens von Bio-Modellen entstand während eines Forschungsaufenthalts des Autors dieser Arbeit in der *Computational Systems Neurobiology Group* am EBI (*European Bioinformatics Institute*, Hinxton/Cambridge, Großbritannien, <http://www.ebi.ac.uk/>). Die Untersuchungen zu den Wissensfacetten von Bio-Modellen hatten gezeigt, dass keine adäquaten formalen Beschreibungsmittel für das Modellverhalten existierten ([KNÜPFER ET AL. 2006](#)). Zeitgleich wurde die Notwendigkeit einer formalen Beschreibung des Modellverhaltens auch von der SBML- und BioModels.net-Community erkannt.

Der Autor dieser Arbeit begann 2007 unter Betreuung von NICOLAS LE NOVÈRE mit der Entwicklung der *Terminology for Description of Dynamics* (TEDDY). Mit dieser Ontologie sollte eine begriffliche Basis für die Beschreibung der Dynamik von Bio-Modellen bereitgestellt werden. Dabei wurde auf die Sprache der Theorie dynamischer Systeme fokussiert, welche für die Analyse des Verhaltens von Differenzialgleichungsmodellen verwendet wird (vgl. [Abschnitt 5.1](#)). Auf Differenzialgleichungen beruhende Bio-Modelle sind vorherrschend in der BioModels Database ([LE NOVÈRE ET AL. 2006](#)) und sind prototypische SBML-Modelle (vgl. [Abschnitt 1.2.2](#)).

Das motivierende Anwendungsszenario (vgl. [Abschnitt 3.3.3](#)) für die Entwicklung von TEDDY bestand darin, übliche Beschreibungen der Dynamik von Bio-Modellen in systembiologischen Publikation zu annotieren. Ausgangspunkt für die Entwicklung von TEDDY war dementsprechend eine Liste von Konzepten, die bei der Beschreibung der Dynamik von Bio-Modellen verwendet werden. Diese Liste wurde aus Publikationen von Bio-Modellen, aus der Diskussion mit Modellierern (insbesondere von der *Computational Systems Neurobiology Group* am EBI, der *Theoretical Biochemistry Group* der Universität Wien, Österreich und der *Biomolecular Signaling and Control Group* der ETH Zürich, Schweiz) sowie

Projektseite	<a href="http://co.mbine.org/standards/teddy">http://co.mbine.org/standards/teddy</a>
SourceForge-Projekt	<a href="http://sourceforge.net/projects/teddyontology/">http://sourceforge.net/projects/teddyontology/</a>
Darstellung via BioPortal	<a href="http://purl.bioontology.org/ontology/TEDDY">http://purl.bioontology.org/ontology/TEDDY</a>
Mailingliste	<a href="http://sourceforge.net/p/teddyontology/mailman/">http://sourceforge.net/p/teddyontology/mailman/</a> ↔ <a href="http://teddyontology-discuss/">teddyontology-discuss/</a>
Wiki	<a href="http://sourceforge.net/apps/mediawiki/teddyontology">http://sourceforge.net/apps/mediawiki/teddyontology</a>
Ticketsystem	<a href="http://sourceforge.net/p/teddyontology/term-requests/">http://sourceforge.net/p/teddyontology/term-requests/</a>
Versionsverwaltung	<a href="http://sourceforge.net/p/teddyontology/code/">http://sourceforge.net/p/teddyontology/code/</a>
erste Release	<a href="http://co.mbine.org/specifications/teddy.rel-2007-06-04">http://co.mbine.org/specifications/teddy.rel-2007-06-04</a>
aktuelle Release	<a href="http://co.mbine.org/specifications/teddy.rel-2014-04-24">http://co.mbine.org/specifications/teddy.rel-2014-04-24</a>
ID-Namensraum	<a href="http://identifiers.org/teddy/">http://identifiers.org/teddy/</a>

Tabelle 6.1.: Wichtige Internetadressen für TEDDY.

aus der Literatur zur Theorie dynamischer Systeme (STROGATZ 2001, JETSCHKE 1989, ELLNER ET AL. 2006, KUZNETSOV 1998 und GUCKENHEIMER ET AL. 1983) erstellt. Weitere Quellen bei der Entwicklung von TEDDY waren COLEY 2003, HALE ET AL. 1991, HUBBARD ET AL. 1995 und JACKSON 1989 sowie die Online-Enzyklopädien PlanetMath (<http://planetmath.org/>) und IZHIKEVICH 2011.

Die so gewonnene Liste diente in zweifacher Hinsicht der Entwicklung von TEDDY: Zum einen legte sie fest, welche Konzepte mindestens in TEDDY enthalten sein müssen. Zum anderen diente sie einer ersten Klassifikation von Konzepten, die bei der Beschreibung der Dynamik von Bio-Modellen eine Rolle spielen. In zahllosen Diskussionen mit Mitentwicklern und Modellierern wurden bestimmte Designentscheidungen getroffen, die die Grundlage für den Aufbau von TEDDY sind (siehe Abschnitt 6.2.3).

Im Juni 2007 wurde die erste Release ([rel-2007-06-04](#)) von TEDDY veröffentlicht. Erste Präsentationen von TEDDY erfolgten im selben Jahr auf verschiedenen Tagungen (LE NOVÈRE ET AL. 2007a, LE NOVÈRE ET AL. 2007b und LE NOVÈRE ET AL. 2007c). Dabei wurde ein zum Akronym „TEDDY“ passendes Logo entworfen (Abbildung 6.7).

TEDDY wird stetig weiterentwickelt und dabei insbesondere das Feedback der Community eingearbeitet. Die aktuelle Release [rel-2014-04-24](#) wurde in Zusammenarbeit mit ANNA ZHUKOVA erstellt. Diese Release (vgl. auch Anhang F) enthält 145 benannte Konzepte (ausgenommen veraltete Konzepte, die als `owl:deprecated` gekennzeichnet sind). Für Details über den Inhalt von TEDDY siehe Abschnitt 6.1.

### 6.2.2. Implementierung von TEDDY

TEDDY wird in OWL spezifiziert. Die Implementierung von TEDDY erfolgt mit Hilfe von Protégé (<http://protege.stanford.edu/>). Die veröffentlichte Release [rel-2014-04-24](#) enthält neben den explizit festgelegten Axiomen auch Schlussfolgerungen, die mittels HermiT (MOTIK ET AL. 2009) gewonnen wurden.

In Tabelle 6.1 sind die wichtigsten Internetadressen für TEDDY zusammengefasst. Im folgenden Text sind die entsprechenden Internetadressen jeweils weggelassen, um die Lesbarkeit zu erhöhen.

Für die (Weiter-)Entwicklung von TEDDY unter Beteiligung der Community gibt es ein entsprechendes Projekt in SourceForge (<http://sourceforge.net/>), in welchem eine

Mailingliste und ein Wiki für Diskussionen über TEDDY genutzt werden. Ein Ticket-system dient der Erfassung von Änderungs- und Erweiterungswünschen für TEDDY. Die Entwicklungsversionen von TEDDY werden mit Hilfe von SVN (*Apache Subversion*, <http://subversion.apache.org/>) verwaltet.

TEDDY ist in dem Ontologie-Portal BioPortal (WHETZEL ET AL. 2011) gelistet. Dort kann die jeweils aktuelle Release von TEDDY in einer benutzerfreundlichen Art visualisiert und durchsucht werden. Als Teil der COMBINE-Initiative finden sich alle wichtigen Informationen zu TEDDY auf der entsprechenden Projektseite (siehe Tabelle 6.1).

### Codierungskonventionen

Für die Codierung von Konzepten in TEDDY gelten folgende Konventionen:

**ID** Jedes Konzept hat einen eindeutigen Identifikator der Form `^TEDDY_\d{7}$` (in Perl-Syntax für reguläre Ausdrücke), d.h. das Wort „TEDDY“ gefolgt von einem Unterstrich und einer siebenstelligen Ziffernfolge. Der Namensraum von TEDDY ist <http://identifiers.org/teddy/>. Die vollständige ID eines Konzeptes ist die IRI (*Internationalized Resource Identifier*), welche durch Anhängen des entsprechenden Identifikators an den Namensraum entsteht. Z.B. wird das Konzept 'TEDDY entity' [`TEDDY_0000000`] durch folgende IRI identifiziert:

[http://identifiers.org/teddy/TEDDY\\_0000000](http://identifiers.org/teddy/TEDDY_0000000).

**Bezeichner** Jedes Konzept besitzt mindestens einen natürlichsprachlichen Bezeichner als `rdfs:label`. Dieser Bezeichner ist englisch (gekennzeichnet durch das Attribut `xml:lang="en"`) und nach folgenden grammatikalischen Vereinbarungen gebildet (diese richten sich im Wesentlichen nach den Namenskonventionen der OBO Foundry, *The Open Biological and Biomedical Ontologies Foundry*, SMITH ET AL. 2007, siehe auch SCHOBBER ET AL. 2009): (1) Konzeptbezeichner sind, soweit möglich, im Singular anzugeben. (2) Konzeptbezeichner sind kleinzuschreiben. Ausnahmen bilden Eigennamen wie etwa „Hopf“ und großgeschriebene Abkürzungen wie etwa „TEDDY“.

**Synonyme** Für Konzepte können Synonyme angegeben werden. Die Angabe erfolgt als `skos:altLabel` mit einem entsprechenden `xml:lang` Attribut. Es können also auch Bezeichner in anderen Sprachen angegeben werden. Für jedes Synonym ist, wenn möglich, eine Referenz anzugeben (als entsprechende Annotation des `skos:altLabel`).

**Definition** Jedes Konzept ist mit einer natürlichsprachlichen englischen Definition zu versehen als `skos:definition` (mit `xml:lang="en"`).

**Referenz** Es ist, soweit möglich, für jedes Konzept eine Referenz als `rdfs:seeAlso` anzugeben. Die Angabe von Referenzen erfolgt idealerweise durch eindeutige Identifikatoren, wie etwa DOI (*Digital Object Identifier*, PASKIN 2010) oder eine ISBN (dies kann mittels des Namensraums <http://identifiers.org/isbn/> geschehen).

**Obsolete** Konzepte, die nicht mehr verwendet werden, sind in den Zweig '`_obsolete`' [`TEDDY_obsolete`] zu verschieben und mittels `owl:deprecated` als veraltet zu kennzeichnen.

Die Entwicklung von **TEDDY** wird durch Versionierung mit Hilfe von **SVN** nachvollziehbar gestaltet. In unregelmäßigen Abständen erfolgt nach Abstimmung unter den Entwicklern eine neue Release von **TEDDY** und wird über die Mailingliste angekündigt. Die Releases werden nach dem Datum ihrer Veröffentlichung benannt: „rel-JJJJ-MM-TT“.

### 6.2.3. Entwicklungsprozess am Beispiel der Stabilitätseigenschaften

Bei der Entwicklung von **TEDDY** mussten zahlreiche Designentscheidungen getroffen werden. Im Rahmen dieser Arbeit ist es nicht möglich, alle getroffenen Designentscheidungen darzulegen und zu begründen. Stattdessen sollen am Beispiel der Stabilitätseigenschaften von Trajektorien drei typische Probleme erläutert werden, die bei den Entscheidungen für eine geeignete Spezifikation entsprechender Konzepte in **TEDDY** aufgetreten sind: das Problem einer uneinheitlichen Terminologie, das Finden einer angemessenen Repräsentation und die Auswahl zu benennender Konzepte. Diese Probleme treten sicherlich bei der Entwicklung vieler anderer Ontologien ebenfalls auf.

#### Terminologisches Wirrwarr

Die Terminologie der Stabilitätseigenschaften ist sehr uneinheitlich, wie eine Betrachtung von vier Standardwerken zeigt (**Tabelle 6.2**). Die in dieser Arbeit (vgl. **Abschnitt 5.1.2**) und in **TEDDY** verwendete Terminologie folgt KUZNETSOV (im Folgenden sind die jeweiligen Bezeichnungen durch Fettschrift hervorgehoben):

„Definition 1.8 An invariant set  $S_0$  is called **stable** if (i) for any sufficiently small neighborhood  $U \supset S_0$  there exists a neighborhood  $V \supset S_0$  such that  $\varphi^t x \in U$  for all  $x \in V$  and all  $t > 0$ ; (ii) there exists a neighborhood  $U_0 \supset S_0$  such that  $\varphi^t x \rightarrow S_0$  for all  $x \in U_0$ , as  $t \rightarrow +\infty$ .

[...]

Property (i) of the definition is called **Lyapunov stability**. If a set  $S_0$  is Lyapunov stable, nearby orbits do not leave its neighborhood. Property (ii) is sometimes called **asymptotic stability**.“ (**KUZNETSOV 1998**, S. 16, Hervorhebung durch den Autor)

Auch STROGATZ betrachte zwei unabhängige Stabilitätseigenschaften, nennt jedoch asymptotisch stabile Mengen „attracting“:

„We say that  $\mathbf{x}^* = \mathbf{0}$  is an **attracting** fixed point [...] all trajectories that start near  $\mathbf{x}^*$  approach it as  $t \rightarrow \infty$ . [...] We say that a fixed point  $\mathbf{x}^*$  is **Liapunov stable** if all trajectories that start sufficiently close to  $\mathbf{x}^*$  remain close to it for all time. [...] If a fixed point is both Liapunov stable and attracting, we'll call it **stable**, or sometimes **asymptotically stable**.“ (**STROGATZ 2001**, S. 128f, Hervorhebung wie im Original)

HALE ET AL. definieren keine unabhängige Eigenschaft der asymptotischen Stabilität. Ljapunow-stabile Mengen werden kurz als „stabil“ bezeichnet. Die Eigenschaft „asymptotically stable“ wird für Mengen verwendet, die sowohl Ljapunow-stabil als auch asymptotisch stabil sind:



Autor	Ljapunow-stabil	asymptotisch stabil	stabil
KUZNETSOV	„Lyapunov stable“	„asymptotically stable“	„stable“
STROGATZ	„Liapunuv stable“	„attracting“	„stable“ oder „asymptotically stable“
HALE ET AL.	„stable“	—	„asymptotically stable“
JETSCHKE	„stabil“	—	„asymptotisch stabil“

Tabelle 6.2.: Begriffe für Stabilitätseigenschaften in drei Standardwerken: [KUZNETSOV 1998](#), [STROGATZ 2001](#), [HALE ET AL. 1991](#) und [JETSCHKE 1989](#). Die Bezeichnungen im Tabellenkopf entsprechen den in dieser Arbeit verwendeten.

„Roughly speaking, an equilibrium point  $\bar{x}$  is **stable** if all solutions near  $\bar{x}$  stay nearby. If, in addition, nearby solutions tend to  $\bar{x}$  as  $t \rightarrow +\infty$ , the  $\bar{x}$  is **asymptotically stable**.“ ([HALE ET AL. 1991](#), S. 16, Hervorhebung durch den Autor)

Auch JETSCHKE führt wie [HALE ET AL. 1991](#) nur zwei Stabilitätsbegriffe ein und benennt eine Ljapunow-stabile Menge einfach „stabil“ und eine zusätzlich asymptotisch stabile Menge „asymptotisch stabil“:

„Definition 1. a) Ein Fixpunkt  $x^0$  heißt **stabil** (gegenüber kleinen Störungen im Ljapunowschen Sinne), wenn es für alle  $\epsilon > 0$  ein  $\delta > 0$  gibt, so daß für die Lösungen  $x(t) = x(t, x(0))$  gilt

$$\|x(t) - x^0\| < \epsilon \quad \forall t \geq 0, \quad \forall x(0) \text{ mit } \|x(0) - x^0\| < \delta$$

(und  $x(t)$  für alle  $t \geq 0$  existiert).

b) Ein stabiler Fixpunkt heißt **asymptotisch stabil**, falls zusätzlich gilt

$$\lim_{t \rightarrow \infty} \|x(t) - x^0\| = 0 \quad \forall x(0) \text{ mit } \|x(0) - x^0\| < \delta.$$

[...]“ ([JETSCHKE 1989](#), S. 42, Hervorhebung im Original kursiv)

### Angemessene Repräsentation

Das zweite Problem betrifft das Finden einer geeigneten Repräsentation. Im Falle der Repräsentation der Stabilität von Fixpunkten und geschlossenen Trajektorien gibt es im Wesentlichen drei Möglichkeiten (nicht alle der folgenden Konzepte finden sich in [TEDDY](#), weshalb auf die Angabe der IDs verzichtet wird):

- (1) Die Definition entsprechender Unterkonzepte in jedem der drei Hauptkonzepte für Trajektorien, z.B. 'asymptotically stable fixed point', 'asymptotically stable limit cycle' und 'strange attractor' (welcher auch asymptotisch stabil ist). Dabei müssen aber gerade die Teile der Definition wiederholt werden, die die Stabilitätseigenschaft festlegen.



- (2) Um diese Wiederholung zu vermeiden, können die Stabilitätseigenschaften separat definiert werden, z.B. durch das Konzept 'asymptotic stability'. Die Definition eines asymptotisch stabilen Fixpunktes kann diese Eigenschaft benutzen:

```
'asymptotically stable fixed point' EquivalentTo:
  'fixed point'
  and (hasCharacteristic some 'asymptotic stability')
```

- (3) Alternativ können entsprechende Trajektorienklassen mit bestimmten Stabilitätseigenschaften definiert werden, etwa das Konzept 'asymptotically stable behaviour', welches als Unterkonzepte 'asymptotically stable fixed point' und 'asymptotically stable limit cycle' hat.

In [Abbildung 6.5](#) sind die drei Möglichkeiten grafisch veranschaulicht.

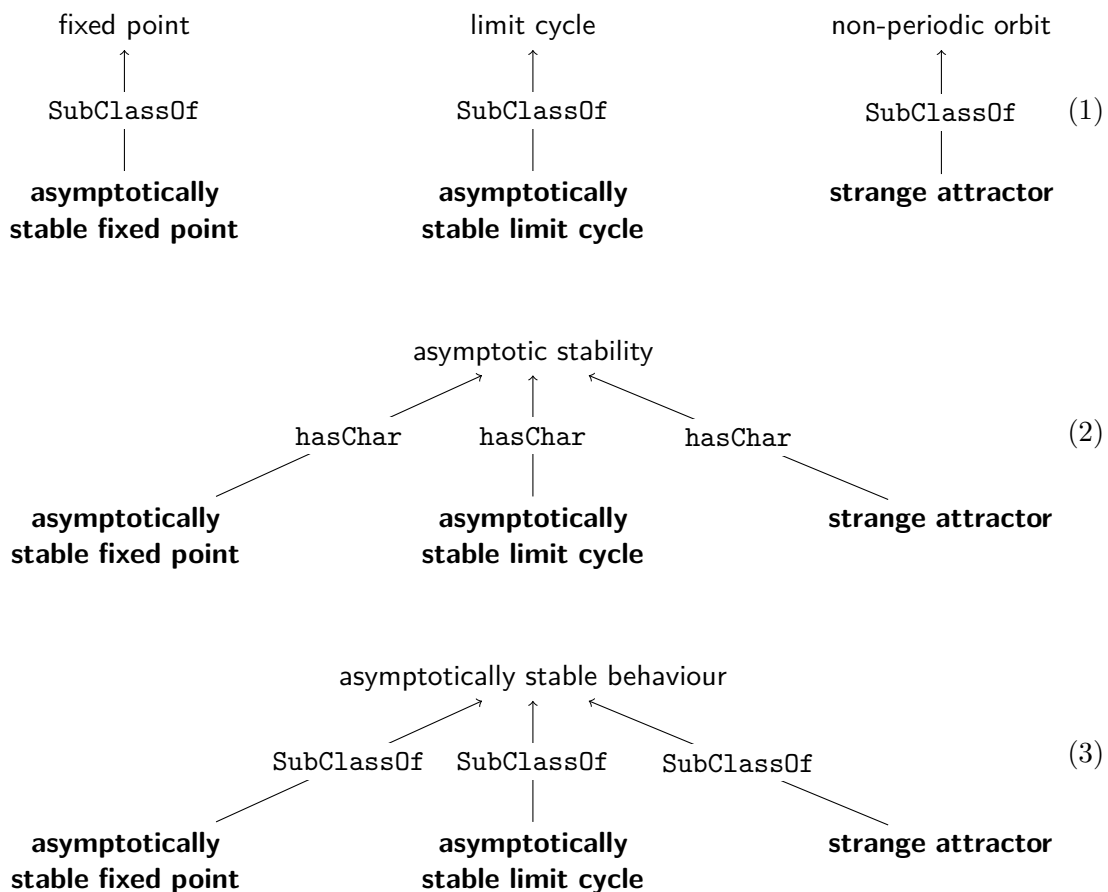


Abbildung 6.5.: Veranschaulichung der im Haupttext beschriebenen [Möglichkeit 1](#) (oben) bis [Möglichkeit 3](#) (unten) der Repräsentation von asymptotisch stabilen Trajektorien (**fett** dargestellt). Dargestellt sind jeweils die Relationen (**SubClassOf** und **hasChar** abkürzend für **hasCharacteristic**), durch welche die asymptotisch stabilen Trajektorien von nicht asymptotisch stabilen Trajektorien unterschieden werden.

Auch [Möglichkeit 2](#) erzeugt, ebenso wie [Möglichkeit 1](#) eine gewisse Redundanz: Neben den Konzepten, die bestimmte Trajektorien betreffen, wie etwa 'asymptotically stable fixed point', werden zusätzlich entsprechende Konzepte für die Stabilitätseigenschaften eingeführt, in diesem Fall etwa 'asymptotic stability'. Die Redundanz liegt hier jedoch nicht in den Konzeptdefinitionen, sondern in der Benennung von Klassen (siehe unten). Zusätzlich ist die Repräsentation in [Möglichkeit 2](#) geringfügig komplexer, da neben der Subsumtion eine weitere Relation in den partiellen Konzeptdefinitionen verwendet wird. Dennoch hat [Möglichkeit 2](#) gegenüber den anderen beiden Möglichkeiten einige Vorteile:

- Die Stabilitätseigenschaften werden als eigenständige Konzepte repräsentiert. Diese können unabhängig von den Konzepten für spezielle Trajektorien benutzt werden (*Wiederverwendbarkeit*).
- In der Theorie dynamischer Systeme werden die Stabilitätseigenschaften als eigenständige Begriffe aufgefasst. [Möglichkeit 2](#) bildet dies in der Ontologie ab (*ontologische Angemessenheit*).
- Die Stabilitätseigenschaften werden in der Theorie dynamischer Systeme für beliebige invariante Mengen definiert. Die [Möglichkeit 1](#) und die [Möglichkeit 3](#) beschränken die Stabilitätseigenschaften jedoch auf Trajektorien (*Allgemeingültigkeit*).
- Die Grundstruktur der Ontologie wird mit [Möglichkeit 2](#) klarer: Es gibt eine Haupthierarchie der Trajektorien (siehe auch unten). [Möglichkeit 3](#) erzeugt neben dieser eine sekundäre Hierarchie der Trajektorien (*Klarheit*).

Aus diesen Gründen ist in [TEDDY Möglichkeit 2](#) umgesetzt. Dabei entsteht jedoch ein weiteres Problem, welches im Folgenden diskutiert werden soll.

### Benennung von Konzepten

Manchmal kann es notwendig sein, über eine individuelle Trajektorie  $x$  nur auszusagen, dass sie asymptotisch stabil ist, ohne festzulegen, ob sie ein Fixpunkt oder eine geschlossene Trajektorie ist. Mit [Möglichkeit 3](#) ist dies leicht umsetzbar (a **Types A** steht für „ $a$  ist eine Instanz von  $A$ “):

```
Individual: x
Types: 'asymptotically stable behaviour'
```

Kann dies auch in [Möglichkeit 2](#) ausgedrückt werden? Durch die Verwendung eines entsprechenden Konzeptausdrucks kann dies realisiert werden:

```
Individual: x
Types: 'temporal behaviour'
      and (hasCharacteristic some 'asymptotic stability')
```

Aus logischer Sicht sind die beiden Aussagen völlig äquivalent. Der ersten Variante liegt folgendes Axiom zu Grunde:

```
Class: 'asymptotically stable behaviour'
EquivalentTo: 'temporal behaviour' and
              (hasCharacteristic some 'asymptotic stability')
```

Ein solches Axiom wird zur Einführung eines benannten Konzeptes („defined concept“, [BAADER ET AL. 2003b](#), S. 56) verwendet. In einer beschreibungslogischen Wissensbasis (ohne zyklische Definitionen) kann jedes Vorkommen eines benannten Konzeptes einfach durch seine Definition ersetzt werden, ohne etwas am logischen Inhalt der codierten Ontologie zu ändern (vgl. [BAADER ET AL. 2003b](#), S. 56ff). Es handelt sich also bei solchen beschreibungslogischen Konzeptdefinitionen um Nominaldefinitionen. Benannte Konzepte sind somit nur Abkürzungen für die sie definierenden Konzeptausdrücke. Die Einführung geeigneter benannter Konzepte ist somit eine Frage der ästhetischen Prägnanz (vgl. [Abschnitt 3.2.3](#)).

Die konkrete Frage ist also, soll bei [Möglichkeit 2](#) zusätzlich ein benanntes Konzept 'asymptotically stable behaviour' eingeführt werden? Allgemein gibt es mit der Einführung benannter Konzepte drei Schwierigkeiten: (a) Die Benennung aller Konzepte ist unmöglich, da deren Anzahl unbeschränkt ist (durch komplexe Klassenausdrücke lassen sich in [OWL](#) beliebig viele anonyme Konzepte bilden, vgl. [Abschnitt 3.3.2](#)). Es sind also möglichst wenige Konzepte zu benennen. (b) Das Finden geeigneter Bezeichner für die Benennung eines Konzeptes. Für Konzepte, die eine Entsprechung in der repräsentierten Domain haben, existieren in der Domainsprache meist geeignete Bezeichner. (c) Die explizite Hierarchie der Ontologie wird umso unübersichtlicher, je mehr benannte Konzepte es gibt. Durch Einführung geeigneter Oberkonzepte kann eine gewisse Ordnung jedoch wieder hergestellt werden. Für die ästhetische Prägnanz einer Ontologie ist es also notwendig, so viele Konzepte wie notwendig und so wenige wie möglich zu benennen.

In [TEDDY](#) wird neben die Konzepte für die Hauptklassifikation von Trajektorien (nach deren Periodenlänge) ein Konzept 'limit behaviour' [[TEDDY\\_0000085](#)] gestellt, das eine sekundäre Klassifikation von Trajektorien nach deren Stabilitätseigenschaften darstellt. Das Konzept 'attractor' [[TEDDY\\_0000094](#)], welches asymptotisch stabile Trajektorien umfasst, findet sich dort als Unterkonzept. Ein wesentlicher Beweggrund für diese Lösung war die Einführung wichtiger Konzepte der Theorie dynamischer Systeme, wie etwa „Attraktor“ und „Repeller“ bei gleichzeitigem Festhalten an [Möglichkeit 2](#), d.h. einer expliziten Repräsentation von Stabilitätseigenschaften. Dafür wird eine zusätzliche Redundanz durch die sekundäre Klassifikation in Kauf genommen.

## 6.3. Verwendung von TEDDY

In diesem Abschnitt sollen Hinweise zur Verwendung von [TEDDY](#) gegeben werden. Dabei wird zunächst gezeigt, wie [TEDDY](#) zur Annotation der Dynamik von Modellen eingesetzt werden kann ([Abschnitt 6.3.1](#)). Dabei zeigt sich, dass für eine umfassende proportionale Beschreibung der Dynamik eine formale Sprache benötigt wird, die auf [TEDDY](#) als Vokabular zurückgreift ([Abschnitt 6.3.2](#)). Abschließend wird diskutiert, inwiefern sich [TEDDY](#) auch für die Charakterisierung experimenteller Daten und approximativer Simulationsergebnisse eignet ([Abschnitt 6.3.3](#)). Dieser Abschnitt hat dabei bewusst Ausblickscharakter, die skizzierten Lösungsansätze konnten im Rahmen dieser Arbeit nicht verwirklicht werden.

### 6.3.1. Annotationen mit TEDDY

[TEDDY](#)-Konzepte können in Annotationen verwendet werden, um das Verhalten von Bio-Modellen zu spezifizieren. Dabei stellen sich zwei grundsätzliche Fragen: Welches

Artefakt wird mit den entsprechenden TEDDY-Annotationen versehen? Was bedeuten diese Annotationen dann jeweils?

Zunächst können konkrete Simulationsresultate, z.B. gegeben in NuML (*Numerical Markup Language*, <http://code.google.com/p/numl/>) mit geeigneten TEDDY-Konzepten annotiert werden. Insbesondere können Zeitreihen mit Unterkonzepten von 'temporal behaviour' [TEDDY\_0000083] qualitativ spezifiziert werden und spezifische Punkte in Parameterscans durch Unterkonzepte von 'bifurcation' [TEDDY\_0000053]. Die Annotationen klassifizieren dabei das zeitliche Verhalten bzw. spezifische Punkte im Parameterraum als zugehörig zum TEDDY-Konzept, auf welches verwiesen wird.

Darüber hinaus können die TEDDY-Annotationen auch den Beschreibungen der Simulationsexperimente zugeordnet werden, die zu einem entsprechenden Modellverhalten führen. So könnten z.B. Beschreibungen von Zeitreihen-Simulationen in SED-ML (*Simulation Experiment Description Markup Language*, WALTERMATH ET AL. 2011b) mit den Unterkonzepten von 'temporal behaviour' [TEDDY\_0000083] annotiert werden. Eine solche Annotation charakterisiert dann das zeitliche Verhalten, das durch die gegebene Simulation des betrachteten Modells hervorgerufen wird.

Schließlich können Bio-Modelle selbst mit TEDDY-Annotationen versehen werden, in SBML (*Systems Biology Markup Language*, HUCKA ET AL. 2003) entspricht das einer Annotation des `model`-Elements. Eine Annotation eines Modells mit einem TEDDY-Konzept ist wie folgt zu interpretieren: Es gibt eine Verwendung des Modells, in der sich ein Verhalten zeigt, welches durch das TEDDY-Konzept charakterisiert werden kann. Dadurch werden also *potenzielle* dynamische Eigenschaften des Modells charakterisiert. Durch entsprechende Qualifikatoren könnte auch spezifiziert werden, dass eine bestimmte dynamische Eigenschaft, die durch eine TEDDY-Annotation gegeben ist, etwa vom Modellierer *intendiert* ist bzw. in der zugehörigen Publikation beschrieben wird. Im folgenden Beispiel 6.2 wird eine solche Qualifikation von TEDDY-Annotationen vorgenommen. Dieses Beispiel präsentiert lediglich erste Ideen zur Annotation des Modellverhaltens. Insbesondere sind geeignete Qualifikatoren erst noch zu entwickeln und deren Bedeutung präzise festzulegen.

### Beispiel 6.2 (Annotation des Verhaltens von Modell 1 mit TEDDY)

Die Simulationsresultate welche in Beispiel 4.5 beschrieben sind (siehe auch Tabelle 2.2), können mit 'stable limit cycle' [TEDDY\_0000114] annotiert werden, genauso, wie die zugehörige Beschreibung der Simulation in SED-ML (als Ergänzung von Zeilen 15 in Anhang E):

```
<task id="task1" modelReference="model1" simulationReference="sim1">
  <annotation>
    <rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:bqdyn="http://biomodels.net/dynamics-qualifiers/"
      <rdf:Description rdf:about="#001">
        <bqdyn:resultsIn>
          <rdf:Bag>
            <rdf:li rdf:resource="http://identifiers.org/
                                  teddy/TEDDY_0000114"/>
          </rdf:Bag>
        </bqdyn:resultsIn>
      </rdf:Description>
    </rdf:RDF>
  </annotation>
</task>
```

Dabei wurde ein fiktiver Qualifikator `resultsIn` aus dem ebenso fiktiven Namensraum `bqdyn` (analog zu den Namensräumen der [BioModels.net Qualifiers](http://co.mbine.org/standards/qualifiers), <http://co.mbine.org/standards/qualifiers>) verwendet, um anzuzeigen, dass das Ergebnis von `task1` qualitativ als stabiler Grenzzzyklus beschrieben werden kann.

Die [SBML](#)-Spezifikation von `Model1 1` ([Anhang C](#)) kann um eine entsprechende Annotation ergänzt werden, welche einen stabilen Grenzzzyklus ('stable limit cycle' [[TEDDY\\_0000114](#)]) als potentiell Verhalten von `Model1 1` angibt (die originalen Kontextzeilen sind ausgegraut und mit Zeilennummern entsprechend [Anhang C](#) versehen):

```

9  <model id="BIOMD0000000005" name="Tyson1991 - Cell Cycle 6 var"
10    metaid="_000001">
59  <annotation>
60    <rdf:RDF
61      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:bqdyn="http://biomodels.net/dynamics-qualifiers/"
67    <rdf:Description rdf:about="#_000001">
        <bqdyn:hasCapability rdf:ID="statement1">
          <rdf:Bag>
            <rdf:li rdf:resource="http://identifiers.org/teddy/TEDDY_0000114"/>
          </rdf:Bag>
        </bqdyn:hasCapability>
131    </rdf:Description>
132  </rdf:RDF>
133 </annotation>

```

Dabei zeigt der Qualifikator `hasCapability` an, dass der stabile Grenzzzyklus ein mögliches Verhalten ist. In [TYSON 1991](#) wird dieser stabile Grenzzzyklus explizit als beobachtetes Verhalten von `Model1 1` im Rahmen der intendierten Verwendung angeführt. Um diesen Umstand zu repräsentieren, kann auf den Mechanismus der Reifikation von [RDF](#)-Aussagen zurückgegriffen werden (siehe [MANOLA ET AL. 2004](#)). Dies ist momentan in [SBML](#) zwar nicht möglich, eine entsprechende Erweiterung ist aber im [SBML Annotation Package](#) ([WALTEMATH ET AL. 2011c](#)) geplant. Die obige Annotation (mit Bezeichner `statement1`) von `Model1 1` könnte selbst wie folgt annotiert werden (vergleiche [WALTEMATH ET AL. 2011c](#)):

```

<rdf:Description rdf:about="#statement1">
  <bqmodel:isDescribedBy>
    <rdf:Bag>
      <rdf:li
        rdf:resource="http://identifiers.org/pubmed/1831270"/>
      </rdf:li>
    </rdf:Bag>
  </bqmodel:isDescribedBy>
</rdf:Description>

```

Damit wird spezifiziert, dass die Aussage über einen stabilen Grenzzzyklus als mögliches Verhalten von `Model1 1` (`statement1`) in der Publikation [TYSON 1991](#) (*PubMed Identifier 1831270*) getroffen wurde. ■

### 6.3.2. DYML: Beschreibungssprache für die Dynamik von Bio-Modellen

Wie bereits in [Abschnitt 4.2.3](#) festgestellt wurde, reicht [TEDDY](#) nicht aus, um die Dynamik eines Bio-Modells ([Facette B3](#)) propositional zu beschreiben. [TEDDY](#) liefert nur das

Vokabular. Für qualitative Aussagen über das Modellverhalten ist eine entsprechende formale Sprache notwendig, die dieses Vokabular verwendet. Es existiert noch keine solche Sprache. **DYML** (*Dynamics Markup Language*, KNÜPFER ET AL. 2013b) ist die Vision einer formalen Sprache für die Dynamik von Bio-Modellen. Damit **DYML** die intrinsische **Facette B3** umfassend formalisieren kann, sollten folgende Anforderungen erfüllt sein:

1. **DYML** kann Phasenportraits qualitativ beschreiben. Das umfasst eine Spezifikation charakterisierender Trajektorien (vgl. **Abschnitt 6.1.2**) sowie Beziehungen zwischen diesen.
2. **DYML** kann Diversifikationen beschreiben. Das erfordert eine Spezifikation verschiedener Phasenportraits sowie der Abhängigkeiten zwischen Phasenportraits.
3. **DYML** verbindet dabei die Phasenportraits einer Diversifikation mit den zugehörigen Bedingungen (Parameter bzw. Systemzustand).
4. **DYML** kann die qualitativen Beschreibungen von Dynamik mit entsprechenden Simulationsdaten in Beziehung setzen.

Für die Verbindung von Phasenportraits zu den zugehörigen Bedingungen sind zwei Ansätze vorstellbar: Einerseits können die Bedingungen innerhalb von **DYML** spezifiziert werden. Dieser Ansatz wird auch im unten stehenden **Beispiel 6.3** skizziert. Andererseits könnten die Bedingungen auch in **SED-ML** spezifiziert werden. Die Verbindung erfolgt dann durch entsprechende Referenzierung.

Im Rahmen dieser Arbeit war es nicht möglich, die Ideen für **DYML** zu verwirklichen. Das folgende **Beispiel 6.3** illustriert die Vision einer Formalisierung der Dynamik von Modell 2.

### **Beispiel 6.3 (Formalisierung der Bifurkation von Modell 2 mit **DYML**)**

In diesem Beispiel wird gezeigt, wie eine Formalisierung der Dynamik von Modell 2 mit Hilfe von **DYML** aussehen könnte. Mit folgendem **DYML**-Ausdruck (**Seite 205**) wird die erste Bifurkation aus **Beispiel 5.2** beschrieben, welches die Änderung der Dynamik von Modell 2 unter Variation des Parameters  $k_6$  darstellt. Dieser **DYML**-Ausdruck lässt sich wie folgt verbalisieren: Durch die superkritische Hopf-Bifurkation **bif1** bei  $k_6 = 0.34341 \text{ min}^{-1}$ , die Modell 2 durchläuft, ändert sich das Phasenportrait **portrait1** zu dem Phasenportrait **portrait2**. Die Bedingungen für das Auftreten eines bestimmten Phasenportraits sind durch **ParameterConstraint** formalisiert. Die Abweichung vom analytisch berechneten Bifurkationspunkt in **Beispiel 5.2** ergibt sich aus den von **TYSON 1991** (S. 7332) verwendeten Approximationen, die Grundlage für die hier verwendeten Constraints sind. Das Phasenportrait am Bifurkationspunkt wird hier zugunsten einer kompakteren Darstellung weggelassen. Dabei ist die Dynamik unterhalb des Bifurkationspunktes (**portrait1**) durch einen stabilen Strudel (**p1f2**) gekennzeichnet (charakteristische Trajektorien werden als **feature** bezeichnet), dem sich alle anderen Trajektorien (**p1f1**) asymptotisch nähern und dabei als gedämpfte Oszillation erscheinen. Die Lage des stabilen Strudels ist durch **StateConstraint** gegeben, entsprechend **Gleichung 5.1** und **Gleichung 5.2** aus **Beispiel 5.1**. Die Bifurkation führt zu **portrait2**, welches durch einen instabilen Strudel (**p2f3**) und einen stabilen Grenzzyklus (**p2f2**) gekennzeichnet ist, dem sich alle anderen Trajektorien (**p2f1**) asymptotisch nähern (als gedämpfte Oszillation).

```

listOfBifurcations
  bifurcation id="bif1"
    bifurcationParameter  $k_6$ 
    bifurcationPoint  $0.34341 \text{ min}^{-1}$ 
    type "TEDDY_0000074" (supercritical hopf bifurcation)
    subPhasePortrait ref="portrait1"
    superPhasePortrait ref="portrait2"
  phasePortrait id="portrait1"
    listOfParameterConstraints
      parameterConstraint  $\frac{k_1[aa]}{k_6[CT]} > \sqrt{\frac{k_6}{k_4}}$ 
    listOfFeatures
      feature id="p1f1"
        listOfStateConstraints
          stateConstraint  $u \neq \frac{k_1[aa]}{k_6[CT]}$ 
          stateConstraint  $v \neq u + \frac{k_6 u}{k_4(\alpha + u^2)}$ 
          type "TEDDY_0000146" (asymptotic behaviour)
          hasCharacteristic "TEDDY_0000063" (damped oscillation)
          limitFeature "p1f2"
        feature id="p1f2"
          listOfStateConstraints
            stateConstraint  $u = \frac{k_1[aa]}{k_6[CT]}$ 
            stateConstraint  $v = u + \frac{k_6 u}{k_4(\alpha + u^2)}$ 
            type "TEDDY_0000126" (stable spiral)
      phasePortrait id="portrait2"
        listOfParameterConstraints
          parameterConstraint  $\sqrt{\alpha} < \frac{k_1[aa]}{k_6[CT]} < \sqrt{\frac{k_6}{k_4}}$ 
        listOfFeatures
          feature id="p2f1"
            listOfStateConstraints
              stateConstraint  $u \neq \frac{k_1[aa]}{k_6[CT]}$ 
              stateConstraint  $v \neq u + \frac{k_6 u}{k_4(\alpha + u^2)}$ 
              type "TEDDY_0000146" (asymptotic behaviour)
              hasCharacteristic "TEDDY_0000063" (damped oscillation)
              limitFeature "p2f2"
            feature id="p2f2"
              type "TEDDY_0000114" (stable limit cycle)
          feature id="p2f3"
            listOfStateConstraints
              stateConstraint  $u = \frac{k_1[aa]}{k_6[CT]}$ 
              stateConstraint  $v = u + \frac{k_6 u}{k_4(\alpha + u^2)}$ 
              type "TEDDY_0000127" (unstable spiral)

```

Dabei sind die Schlüsselwörter von **DYML** **fett** gedruckt. Die Constraints sind der Phasenraumanalyse in [TYSON 1991](#) (S. 7331f) entnommen. Auf die Darstellung des Bezuges zu Simulationsdaten wird in diesem Beispiel verzichtet.

Diese Charakterisierung der Phasenportraits unter- und oberhalb des Bifurkationspunktes ist konsistent mit den folgenden **TEDDY**-Axiomen für die superkritische Hopf-Bifurkation ('supercritical Hopf bifurcation' [[TEDDY\\_0000074](#)]) entsprechend [Beispiel 6.1](#):

```

Class: 'supercritical Hopf bifurcation'
SubClassOf: (below some 'stable spiral')
             and (above some 'unstable spiral')
             and (above some 'stable limit cycle')

```



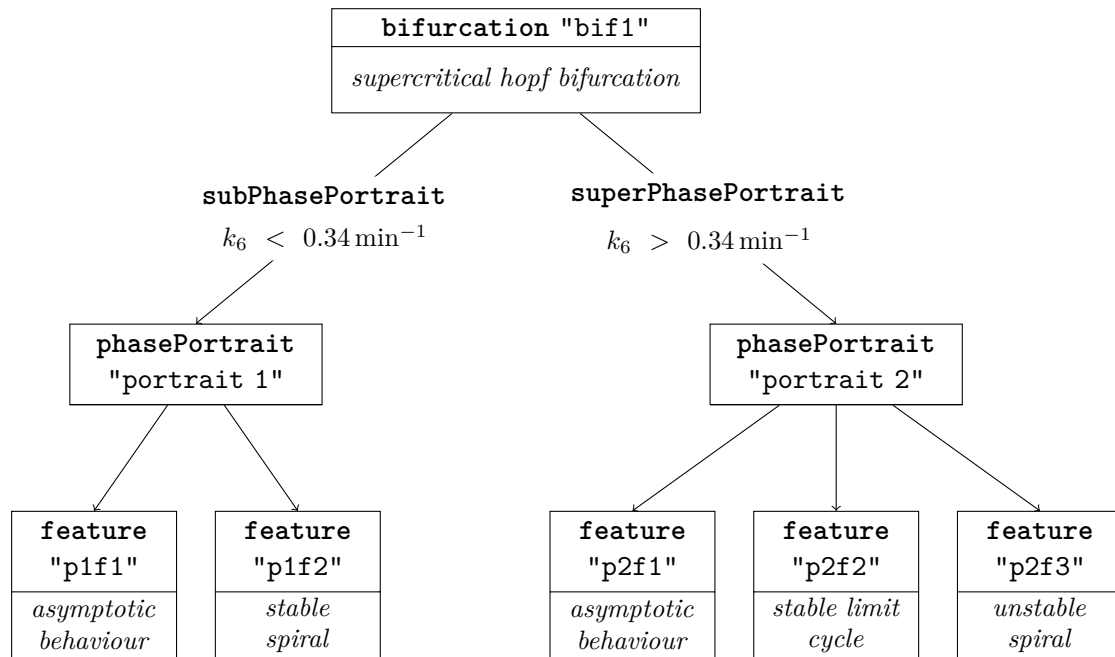


Abbildung 6.6.: Wichtige Elemente der **DYML**-Beschreibung einer Bifurkation von Modell 2 aus Beispiel 6.3. Als **type** zugeordnete **TEDDY**-Konzepte sind *kursiv* dargestellt. Die oberen beiden Kanten stellen Zuordnungen entsprechender Phasenportraits zur Bifurkation mit den zugehörigen Bedingungen dar, die unteren Kanten ordnen den beiden Phasenportraits jeweils sie charakterisierende Trajektorien zu. Für Details siehe Text.

Die obige **DYML**-Beschreibung ist jedoch wesentlich detailreicher als die **TEDDY**-Formalisierung.

In [Abbildung 6.6](#) sind die wichtigsten Elemente obenstehender **DYML**-Beschreibung und ihre Beziehungen untereinander illustriert. ■

### 6.3.3. Repräsentation realen Verhaltens

Die von **TEDDY** beschriebenen Konzepte der Theorie dynamischer Systeme sind präzise definierte, idealisierte mathematische Begriffe. Diese treffen natürlich nicht auf verrauschte, fehlerbehaftete und unvollständige Beobachtungen von Wirklichkeitsausschnitten zu. Genau genommen sind diese idealisierten Begriffe auch nicht für Simulationsresultate verwendbar, die im Allgemeinen ja nur Approximationen des tatsächlichen Verhaltens eines dynamischen Systems darstellen. Besonders deutlich wird dies bei stochastischen Simulationen. Aber auch bei deterministischen Simulationen werden bereits durch die endliche Genauigkeit der Zahlendarstellung im Rechner Näherungen notwendig. Allerdings gestatten sowohl Simulationsresultate als auch experimentelle Beobachtungen Rückschlüsse auf die Dynamik des idealisierten dynamischen bzw. natürlichen Systems. **TEDDY** kann eingesetzt werden, um die Dynamik von Bio-Modellen auf der Grundlage solcher Interpretationen von experimentellen Beobachtungen und Simulationsresultaten zu beschreiben. Dafür müssen

geeignete Qualifikatoren definiert werden, die die Art der Interpretation charakterisieren. So könnte etwa durch `looksLike 'stable fixed point'` ausgedrückt werden, dass eine experimentell beobachtete Zeitreihe so „aussieht“, als ob sich das System in einem stabilen Fixpunkt befindet (vgl. auch die Aussagen zu phänomenologischen Beschreibungen zeitlichen Verhaltens in [Abschnitt 5.1.2](#)).

Basiert die Beschreibung der Dynamik eines Systems auf der Interpretation von Daten aus Simulationen oder experimentellen Messungen, ist es wichtig, diese Interpretation nachvollziehbar zu machen. Das heißt insbesondere, dass für Aussagen über die Dynamik eines Systems dokumentiert wird, wie sie zustande gekommen sind, welche Annahmen dabei zugrunde gelegt wurden und wie zuverlässig sie sind. Es gibt Versuche, solches Metawissen über die Herkunft und Zuverlässigkeit von Aussagen zu formalisieren, etwa die [ECO](#) (*Evidence Codes Ontology*, <http://www.evidenceontology.org>), welche Belege für Aussagen klassifiziert und [ORCA](#) (*Ontology of Reasoning, Certainty and Attribution*, [DE WAARD ET AL. 2012](#)), einer Ontologie für die Bewertung der Zuverlässigkeit von Aussagen. [ECO](#) ist auf Aussagen über die Funktionen von Genen spezialisiert und hat keine geeigneten Terme für die Art von Aussagen über die Dynamik von Bio-Modellen. [ORCA](#) ist nicht ausreichend entwickelt, um für die Annotation von Aussagen über die Dynamik von Modellen hilfreich zu sein. Neben den oben angesprochenen Qualifikatoren müssten also auch geeignete Mittel entwickelt werden, dieses Metawissen über die Annotationen der Dynamik zu repräsentieren.

## Zusammenfassung

In diesem Kapitel wurde **TEDDY** vorgestellt. **TEDDY** ist eine Ontologie, die begriffliches Wissen der Theorie dynamischer Systeme formalisiert ([Abschnitt 6.1](#)). Es wurde gezeigt, wie **TEDDY** als Community-Projekt entwickelt wurde und weiter entwickelt wird ([Abschnitt 6.2](#)). Abschließend wurden Aspekte der Verwendung von **TEDDY** beleuchtet ([Abschnitt 6.3](#)). [Abbildung 6.7](#) zeigt das vom Autor dieser Arbeit entworfenen Logo für **TEDDY**.

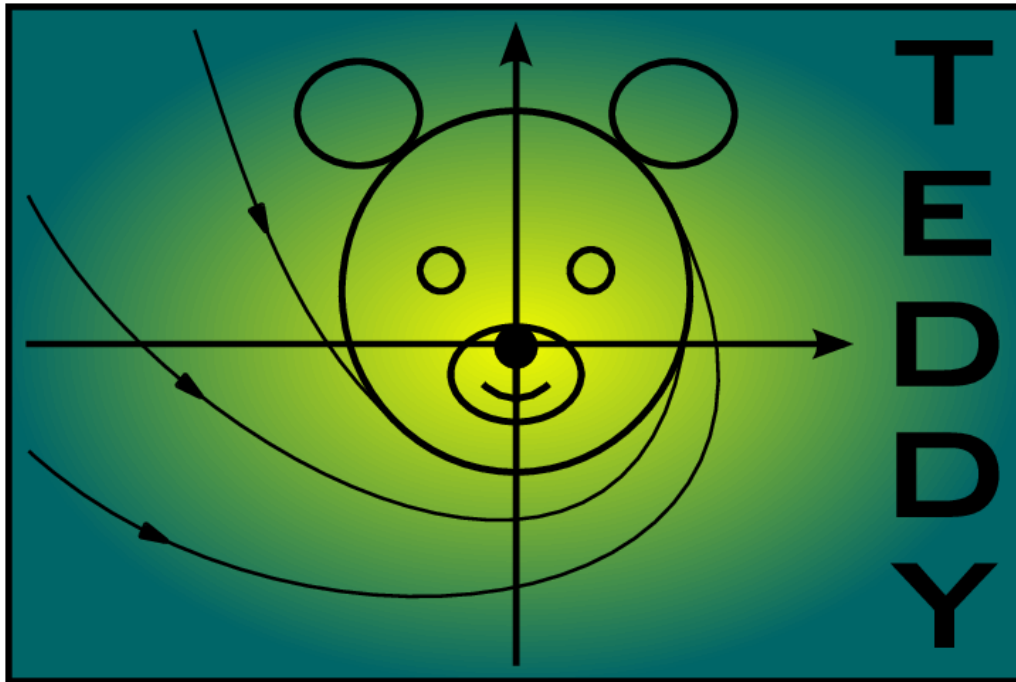


Abbildung 6.7.: Logo für **TEDDY**. Das Logo deutet einen stabilen Grenzyklus im Phasenraum an, welcher einen „Teddy“ umrahmt. Entworfen wurde dieses Logo vom Autor dieser Arbeit im Juni 2007

Ein Schwerpunkt im Zusammenhang mit der zukünftigen Entwicklung von **TEDDY** sollte auf der Verwendung von **TEDDY** zur Annotation des Verhaltens von Modellen bzw. realen Systemen sowie der Entwicklung formaler Sprachen liegen, die auf **TEDDY** als Vokabular zurückgreifen (wie etwa **DYML**). Erst die Anwendung von **TEDDY** führt zu einer zielgerichteten Weiterentwicklung der Ontologie. Idealerweise könnten qualitative Techniken, wie sie in [Kapitel 5](#) vorgestellt wurden, benutzt werden, um das Verhalten von Bio-Modellen automatisch mit **DYML** und **TEDDY** zu charakterisieren. Außerdem könnten Methoden der Mustererkennung auf die Simulationsresultate für denselben Zweck angewendet werden.

Zum Abschluss dieser Arbeit wurde in diesem Kapitel durch **TEDDY** demonstriert, wie das begriffliche Wissen über das dynamische Verhalten von Modellen formal repräsentiert werden kann. Damit ist bezüglich der Dynamik von Bio-Modellen eine wichtige Grundlage für die computergestützte Modellierung und Simulation in der Systembiologie gelegt worden.

# Zusammenfassung und Ausblick

„Model-based simulation is like a gem: it is multifaceted.“

---

(ZEIGLER 1984: *Multifaceted modelling and discrete event simulation*, S. v)

Das Verstehen komplexer biologischer Phänomene auf der Basis dynamischer Modelle ist ohne Computerunterstützung nicht möglich. Seit mehr als einem Jahrzehnt werden enorme Anstrengungen unternommen, eine Softwareinfrastruktur für die systembiologische Modellierung und Simulation zu entwickeln.

Die vorliegende Arbeit bietet den dafür notwendigen Standards, formalen Sprachen, Ontologien und auf diesen operierenden Programmen ein wissenschaftstheoretisches Fundament in Gestalt der Wissensfacetten von Bio-Modellen. Die Wissensfacetten von Bio-Modellen bieten einen systematischen Zugang zu der zentralen Frage dieser Arbeit: Welches Wissen muss in welcher Form formal repräsentiert sein, um als Grundlage für eine allumfassende computergestützte Modellierung und Simulation in der Systembiologie dienen zu können?

Die Wissensfacetten von Bio-Modellen (Kapitel 2) stellen ein konzeptuelles Schema zur systematischen Beschreibung von Bio-Modellen, ihrer Verwendung und ihres Verhaltens dar. Dabei muss eine vollständige Beschreibung eines Bio-Modells seine Struktur, seine Funktion und sein Verhalten umfassen und die intrinsische formale Semantik mit der extrinsischen biologischen Wirklichkeit in Beziehung setzen.

Die formale Repräsentation der Wissensfacetten von Bio-Modellen bietet eine Grundlage für eine umfassende Computerunterstützung der Modellierung und Simulation in der Systembiologie. Geleitet durch die Wissensfacetten wurden existierende Ansätze für die formale Repräsentation relevanter Wissensfragmente eingeordnet und bewertet sowie Defizite an Beschreibungsmitteln identifiziert (Kapitel 4). Das Fehlen eines Ansatzes zur formalen Repräsentation der teleologischen Funktion als zentralem Treiber der Modellierung und Simulation schränkt die Computerunterstützung bei der Auswahl, Kombination und Spezifikation von Methoden der Modellverwendung stark ein. In dieser Arbeit wurde mit MINTENTO (*Modelling Intention Ontology*, KNÜPFER ET AL. 2013a) eine Ontologie skizziert, die helfen kann, dieses Defizit zu beseitigen.

Ein größeres Defizit liegt im Fehlen formalsprachlicher Mittel zur qualitativen Beschreibung des Verhaltens von Bio-Modellen. Im Rahmen dieser Arbeit wurde die Ontologie TEDDY (*TErminology for the Description of DYnamics*, COURTOT ET AL. 2011) entwickelt (siehe Kapitel 6), auf deren Grundlage sich das Verhalten von Bio-Modellen propositionell formalisieren lässt. TEDDY stellt eine formale begriffliche Basis der Theorie dynamischer Systeme dar.

Die drei Hauptresultate der vorliegenden Arbeit – die Wissensfacetten von Bio-Modellen (in Teil I), das Review formaler Repräsentationsmittel für die Wissensfacetten von Bio-Modellen (in Teil II) und TEDDY (in Teil III) – sind eingebettet in grundsätzliche

Betrachtungen zur Modellierung und Simulation in der Systembiologie ([Kapitel 1](#)) und basieren auf allgemeinen Begriffen und Verfahren der Wissensrepräsentation ([Kapitel 3](#)) sowie speziellen Methoden der qualitativen Beschreibung von Dynamik ([Kapitel 5](#)). Die wichtigsten Resultate und theoretischen Wissensgebiete der vorliegenden Arbeit sind in [Abbildung 7.1](#) im Wissensdiamanten dargestellt (vgl. [Abbildung 0.1](#)).

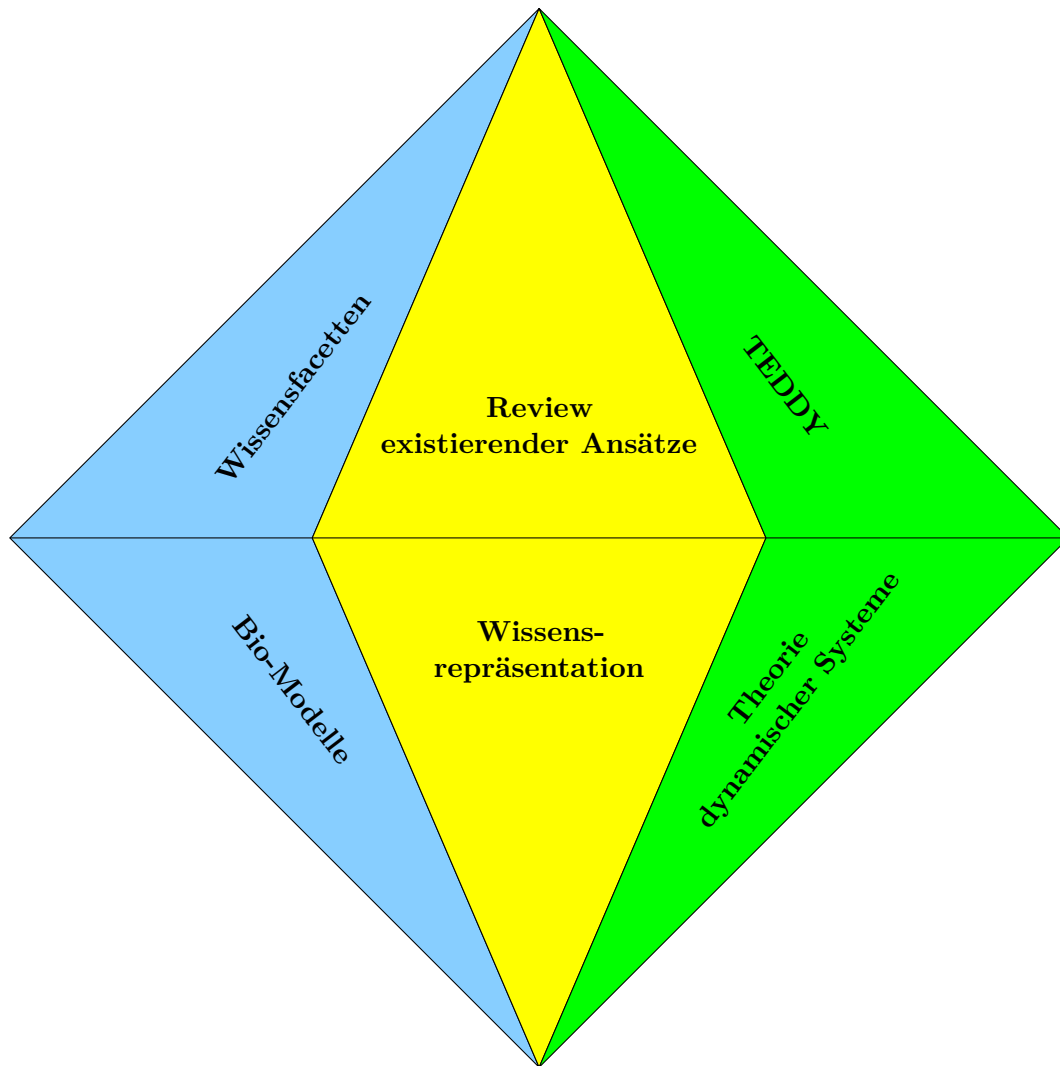


Abbildung 7.1.: Wichtige Resultate (oben) und theoretische Grundlagen (unten) der vorliegenden Dissertation dargestellt im Wissensdiamanten (siehe [Abbildung 2.7](#)).

Die vorliegende Arbeit bietet mit den Wissensfacetten von Bio-Modellen ein konzeptuelles Schema für die formale Repräsentation relevanter Wissensfragmente. Entsprechende formale Sprachen, Ontologien und auf diese aufbauende Algorithmen gilt es jedoch noch zu entwickeln bzw. weiterzuentwickeln. Als ferne Ziele gelten dabei sicherlich eine umfassende und reichhaltige formale Repräsentation der modellierten biologischen Wirklichkeit und die Entwicklung „intelligenter“ Werkzeuge für die Modellierung und Simulation biologischer Phänomene, die dieses formal repräsentierte Wissen nutzen. Beide Ziele sind sicherlich

---

nicht in den nächsten Jahren realisierbar.

Konkrete nächste Ziele bestehen in der Umsetzung der in dieser Arbeit nur umrissenen Repräsentationsmittel, der Ontologie [MINTENTO](#) für die teleologische Funktion von Bio-Modellen und der formalen Sprache [DYML](#) (*Dynamics Markup Language*, [KNÜPFER ET AL. 2013b](#)) für die Beschreibung der Dynamik von Bio-Modellen. Außerdem schließen sich an diese Arbeit zwei Forschungsthemen an, die in eigenständigen Projekten beantwortet werden könnten. Zum einen wären Verfahren wünschenswert, welche die Dynamik eines Bio-Modells automatisch charakterisieren und entsprechende formale qualitative Beschreibung des Verhaltens generieren. Zum anderen erscheint es durchaus plausibel, die Ergebnisse dieser Arbeit auf andere Forschungsgebiete auszudehnen, in denen dynamische Modellierung eine Rolle spielt. Dafür müssten die Wissensfacetten entsprechend verallgemeinert werden und die existierenden Ansätze in den anderen Forschungsgebieten entsprechend bewertet und eingeordnet werden.

Die vorliegende Arbeit bietet nicht nur die konzeptuellen Grundlagen für die Entwicklung einer auf formaler Wissensrepräsentation basierenden umfassenden Computerunterstützung der Modellierung und Simulation, sie möchte auch ein tiefergehendes Verstehen der Prozesse der Modellierung und Simulation erreichen, also neben Performanz auch Kompetenz bieten. Dafür wurde der Wissensdiamant von allen Seiten betrachtet und die Struktur, die Funktion und das Verhalten dynamischer Modelle der Systembiologie beleuchtet.





# Literatur

- ABELSON, H. und SUSSMAN, G. J. (1996). *Structure and Interpretation of Computer Programs*. 2. Aufl. Cambridge, MA, USA: MIT Press. ISBN: 978-0-262-01153-2.
- ALON, U. (2007a). *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Mathematical and Computational Biology Series. Chapman und Hall/CRC. ISBN: 978-1-58488-642-6.
- ALON, U. (2007b). *Network motifs: theory and experimental approaches*. In: *Nature Reviews Genetics* 8:6 (Juni 2007), S. 450–461. DOI: [10.1038/nrg2102](https://doi.org/10.1038/nrg2102).
- APT, K. (2003). *Principles of Constraint Programming*. Cambridge University Press. ISBN: 978-0-521-82583-2.
- APT, K. R. und BRAND, S. (2005). *Constraint-Based Qualitative Simulation*. In: *12th International Symposium on Temporal Representation and Reasoning (TIME 2005)*. IEEE Computer Society, S. 26–34. DOI: [10.1109/TIME.2005.19](https://doi.org/10.1109/TIME.2005.19).
- APWEILER, R.; BAIROCH, A.; WU, C. H.; BARKER, W. C.; BOECKMANN, B.; FERRO, S.; GASTEIGER, E.; HUANG, H.; LOPEZ, R.; MAGRANE, M.; MARTIN, M. J.; NATALE, D. A.; O'DONOVAN, C.; REDASCHI, N. und YEH, L. S. (2004). *UniProt: the Universal Protein knowledgebase*. In: *Nucleic Acids Research* 32:D1 (Jan. 2004): *Database issue*, S. 115–119. DOI: [10.1093/nar/gkh131](https://doi.org/10.1093/nar/gkh131).
- ARISTOTELES (1998). *Die Kategorien*. Ditzingen: Reclam. ISBN: 978-3-15-009706-9.
- AUSBROOKS, R.; BUSWELL, S.; DALMAS, S.; DEVITT, S.; DIAZ, A.; HUNTER, R.; SMITH, B.; SOIFFER, N.; SUTOR, R. und WATT, S. (2003). *Mathematical Markup Language (MathML) Version 2.0*. W3C Recommendation. World Wide Web Consortium. 21. Okt. 2003. URL: <http://www.w3.org/TR/2003/REC-MathML2-20031021/>.
- BAADER, F.; CALVANESE, D.; MCGUINNESS, D.; NARDI, D. und PATEL-SCHNEIDER, P., Hrsg. (2003a). *The Description Logic Handbook - Theory, Implementation and Applications*. New York, NY, USA: Cambridge University Press. ISBN: 978-0-521-78176-3.
- BAADER, F. und NUTT, W. (2003b). *Basic Description Logics*. In: *The Description Logic Handbook - Theory, Implementation and Applications*. Hrsg. von F. BAADER; D. CALVANESE; D. MCGUINNESS; D. NARDI und P. PATEL-SCHNEIDER. New York, NY, USA: Cambridge University Press. Kap. 2, S. 43–95.
- BAIER, C. und KATOEN, J. (2008). *Principles of Model Checking*. MIT Press. ISBN: 978-0-262-02649-9.
- BARD, J.; RHEE, S. Y. und ASHBURNER, M. (2005). *An ontology for cell types*. In: *Genome Biology* 6:2, R21. DOI: [10.1186/gb-2005-6-2-r21](https://doi.org/10.1186/gb-2005-6-2-r21).
- BECKETT, D. (2004). *RDF/XML Syntax Specification (Revised)*. W3C Recommendation. World Wide Web Consortium. 10. Feb. 2004. URL: <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>.
- BECKETT, D.; BERNERS-LEE, T.; PRUD'HOMMEAUX, E. und CAROTHERS, G. (2013). *Turtle – Terse RDF Triple Language*. Hrsg. von G. CAROTHERS und E. PRUD'HOMMEAUX.

- W3C Candidate Recommendation. World Wide Web Consortium. 19. Feb. 2013. URL: <http://www.w3.org/TR/2013/CR-turtle-20130219/>.
- BECKSTEIN, C.; KNÜPFER, C. und KÖNIG-RIES, B. (2011). *Computer-Supported Research – Konzeptpapier zum CoSRe-Projekt*. Techn. Ber. Dokument 21436. Digitale Bibliothek Thüringen. URL: <http://www.db-thueringen.de/servlets/DocumentServlet?id=21436>.
- BENSON, D. A.; CAVANAUGH, M.; CLARK, K.; KARSCH-MIZRACHI, I.; LIPMAN, D. J.; OSTELL, J. und SAYERS, E. W. (2013). *GenBank*. In: *Nucleic Acids Research* 41:D1, S. D36–D42. DOI: [10.1093/nar/gks1195](https://doi.org/10.1093/nar/gks1195).
- BERGLUND, A.; BOAG, S.; CHAMBERLIN, D.; FERNÁNDEZ, M. F.; KAY, M.; ROBIE, J. und SIMÉON, J., Hrsg. (2010). *XML Path Language (XPath) 2.0 (Second Edition)*. W3C Recommendation. World Wide Web Consortium. 14. Dez. 2010. URL: <http://www.w3.org/TR/2010/REC-xpath20-20101214/>.
- BERGMANN, F. T. (2012). *SED-ML: Nested Simulation Proposal*. Version 3. COMBINE. Juni 2012. URL: <http://identifiers.org/combine.specifications/sed-ml.proposal.nested-simulations.FB.version-3>.
- BIOMODELS DATABASE (2014). *28th BioModels Database Release*. 16. Sep. 2014. URL: [http://www.ebi.ac.uk/biomodels-main/static-pages.do?page=release\\_20140916](http://www.ebi.ac.uk/biomodels-main/static-pages.do?page=release_20140916) (besucht am 14.10.2014).
- BOBROW, D. (1985). *Qualitative Reasoning about Physical Systems*. Elsevier Science. ISBN: 978-0-444-59921-6.
- BORNSTEIN, B. J.; KEATING, S. M.; JOURAKU, A. und HUCKA, M. (2008). *LibSBML: an API Library for SBML*. In: *Bioinformatics* 24:6, S. 880–881. DOI: [10.1093/bioinformatics/btn051](https://doi.org/10.1093/bioinformatics/btn051).
- BORST, P.; AKKERMANS, H. und TOP, J. (1997). *Engineering ontologies*. In: *International Journal of Human-Computer Studies* 46:2, S. 365–406. DOI: [10.1006/ijhc.1996.0096](https://doi.org/10.1006/ijhc.1996.0096).
- BRACHMAN, R. und LEVESQUE, H. (2004). *Knowledge Representation and Reasoning*. Morgan Kaufmann. ISBN: 978-1-55860-932-7.
- BRAY, T.; PAOLI, J.; SPERBERG-MCQUEEN, C. M.; MALER, E.; YERGEAU, F. und COWAN, J., Hrsg. (2006). *Extensible Markup Language (XML) 1.1 (Second Edition)*. W3C Recommendation. World Wide Web Consortium. 16. Aug. 2006. URL: <http://www.w3.org/TR/2006/REC-xml11-20060816>.
- BRAZMA, A.; HINGAMP, P.; QUACKENBUSH, J.; SHERLOCK, G.; SPELLMAN, P.; STOECKERT, C.; AACH, J.; ANSORGE, W.; BALL, C. A.; CAUSTON, H. C.; GAASTERLAND, T.; GLENNISON, P.; HOLSTEGE, F. C.; KIM, I. F.; MARKOWITZ, V.; MATESE, J. C.; PARKINSON, H.; ROBINSON, A.; SARKANS, U.; SCHULZE-KREMER, S.; STEWART, J.; TAYLOR, R.; VILO, J. und VINGRON, M. (2001). *Minimum information about a microarray experiment (MIAME) – toward standards for microarray data*. In: *Nature Genetics* 29:4 (Dez. 2001), S. 365–371. DOI: [10.1038/ng1201-365](https://doi.org/10.1038/ng1201-365).
- BREDEWEG, B.; LINNEBANK, F.; BOUWER, A. und LIEM, J. (2009). *Garp3 — Workbench for qualitative modelling and simulation*. In: *Ecological Informatics* 4:5–6: Special Issue: *Qualitative models of ecological systems*, S. 263–281. DOI: [10.1016/j.ecoinf.2009.09.009](https://doi.org/10.1016/j.ecoinf.2009.09.009).
- BRUGGEMAN, F. J. und WESTERHOFF, H. V. (2007). *The nature of systems biology*. In: *Trends in Microbiology* 15:1 (Jan. 2007), S. 45–50. DOI: [10.1016/j.tim.2006.11.003](https://doi.org/10.1016/j.tim.2006.11.003).

- BUREK, P.; HOEHNDOERF, R.; LOEBE, F.; VISAGIE, J.; HERRE, H. und KELSO, J. (2006). *A top-level ontology of functions and its application in the Open Biomedical Ontologies*. In: *Bioinformatics* 22:14, e66–e73. DOI: [10.1093/bioinformatics/btl266](https://doi.org/10.1093/bioinformatics/btl266).
- BUTLER, D. (2012). *Scientists: your number is up*. In: *Nature* 485:7400 (31. Mai 2012), S. 564. DOI: [10.1038/485564a](https://doi.org/10.1038/485564a).
- CELLIER, F. und FISCHLIN, A. (1982). *Computer-assisted modeling of ill-defined systems*. In: *Progress in cybernetics and systems research* 8, S. 417–429.
- CEUSTERS, W.; SMITH, B. und FLANAGAN, J. (2003). *Ontology and Medical Terminology: Why Description Logics Are Not Enough*. In: *Towards an Electronic Patient Record (TEPR 2003)*. URL: <http://ontology.buffalo.edu/medo/TEPR2003.pdf>.
- CHABRIER-RIVIER, N.; CHIAVERINI, M.; DANOS, V.; FAGES, F. und SCHÄCHTER, V. (2004). *Modeling and querying biomolecular interaction networks*. In: *Theoretical Computer Science* 325:1, S. 25–44. DOI: [10.1016/j.tcs.2004.03.063](https://doi.org/10.1016/j.tcs.2004.03.063).
- CHABRIER-RIVIER, N.; FAGES, F. und SOLIMAN, S. (2005). *The Biochemical Abstract Machine BIOCHAM*. In: *Computational Methods in Systems Biology (CMSB 2004)*. International Conference CMSB 2004. Hrsg. von V. DANOS und V. SCHÄCHTER. Lecture Notes in Computer Science 3082. Revised Selected Papers. Berlin, Heidelberg: Springer-Verlag, S. 172–191. DOI: [10.1007/978-3-540-25974-9\\_14](https://doi.org/10.1007/978-3-540-25974-9_14).
- CHANDRASEKARAN, B.; JOSEPHSON, J. R. und BENJAMINS, V. R. (1999). *What Are Ontologies, and Why Do We Need Them?* In: *IEEE Intelligent Systems* 14:1, S. 20–26.
- CHELLIAH, V.; ENDLER, L.; JUTY, N.; LAIBE, C.; LI, C.; RODRIGUEZ, N. und LE NOVÈRE, N. (2009). *Data Integration and Semantic Enrichment of Systems Biology Models and Simulations*. In: *Data Integration in the Life Sciences*, S. 5–15. DOI: [10.1007/978-3-642-02879-3\\_2](https://doi.org/10.1007/978-3-642-02879-3_2).
- CHOMSKY, N. (1965). *Aspects of the Theory of Syntax*. M.I.T. Press. ISBN: 978-0-262-53007-1.
- CHURCH, A. (1958). *Ontological Commitment*. In: *The Journal of Philosophy* 55:23, S. 1008–1014. URL: <http://www.jstor.org/stable/2021909>.
- CIMINO, J. (1996). *Formal descriptions and adaptive mechanisms for changes in controlled medical vocabularies*. In: *Methods of Information in Medicine* 35:3 (Sep. 1996), S. 202–210.
- CIMINO, J. (1998). *Desiderata for controlled medical vocabularies in the twenty-first century*. In: *Methods of Information in Medicine* 37:4-5 (Nov. 1998), S. 394–403.
- CLARKE, E. M.; GRUMBERG, O. und PELED, D. A. (1999). *Model checking*. MIT Press. ISBN: 978-0-262-03270-4.
- COCCHIARELLA, N. (2007). *Formal Ontology and Conceptual Realism*. Formal Ontology and Conceptual Realism. Springer London, Limited. ISBN: 978-1-4020-6204-9.
- COLEY, A. (2003). *Dynamical Systems and Cosmology*. Astrophysics and Space Science Library 291. Kluwer Academic Publishers. ISBN: 978-1-4020-1403-1.
- COOK, D. L.; MEJINO, J. L.; NEAL, M. L. und GENNARI, J. H. (2008). *Bridging biological ontologies and biosimulation: the ontology of physics for biology*. In: *AMIA Annual Symposium Proceedings*, S. 136–140. URL: <http://www.ncbi.nlm.nih.gov/pubmed/18999003>.
- COOPER, L.; WALLS, R. L.; ELSE, J.; GANDOLFO, M. A.; STEVENSON, D. W.; SMITH, B.; PREECE, J.; ATHREYA, B.; MUNGALL, C. J.; RENSING, S.; HISS, M.; LANG, D.; RESKI, R.; BERARDINI, T. Z.; LI, D.; HUALA, E.; SCHAEFFER, M.; MENDA, N.; ARNAUD, E.;

- SHRESTHA, R.; YAMAZAKI, Y. und JAISWAL, P. (2013). *The Plant Ontology as a Tool for Comparative Plant Anatomy and Genomic Analyses*. In: *Plant and Cell Physiology* 54:2, e1. DOI: [10.1093/pcp/pcs163](https://doi.org/10.1093/pcp/pcs163).
- CORCHO, O.; FERNÁNDEZ-LÓPEZ, M. und GÓMEZ-PÉREZ, A. (2003). *Methodologies, tools and languages for building ontologies: where is their meeting point?* In: *Data & Knowledge Engineering* 46:1, S. 41–64. DOI: [10.1016/S0169-023X\(02\)00195-7](https://doi.org/10.1016/S0169-023X(02)00195-7).
- COURTOT, M.; JUTY, N.; KNÜPFER, C.; WALTEMATH, D.; ZHUKOVA, A.; DRÄGER, A.; DUMONTIER, M.; FINNEY, A.; GOLEBIEWSKI, M.; HASTINGS, J.; HOOPS, S.; KEATING, S.; KELL, D. B.; KERRIEN, S.; LAWSON, J.; LISTER, A.; LU, J.; MACHNE, R.; MENDES, P.; POCOCK, M.; RODRIGUEZ, N.; VILLEGIER, A.; WILKINSON, D. J.; WIMALARATNE, S.; LAIBE, C.; HUCKA, M. und LE NOVÈRE, N. (2011). *Controlled Vocabularies and Semantics in Systems Biology*. In: *Molecular Systems Biology* 7, 543 (Okt. 2011). DOI: [10.1038/msb.2011.77](https://doi.org/10.1038/msb.2011.77).
- CREATIVE COMMONS (2009). *CC0 1.0 Universal – Public Domain Dedication*. URL: <https://creativecommons.org/publicdomain/zero/1.0/> (besucht am 22.09.2012).
- CUELLAR, A. A.; LLOYD, C. M.; NIELSEN, P. F.; BULLIVANT, D. P.; NICKERSON, D. P. und HUNTER, P. J. (2003). *An Overview of CellML 1.1, a Biological Model Description Language*. In: *SIMULATION* 79:12, S. 740–747. DOI: [10.1177/0037549703040939](https://doi.org/10.1177/0037549703040939).
- CZAUDERNA, T.; KLUKAS, C. und SCHREIBER, F. (2010). *Editing, validating and translating of SBGN maps*. In: *Bioinformatics* 26:18, S. 2340–2341. DOI: [10.1093/bioinformatics/btq407](https://doi.org/10.1093/bioinformatics/btq407).
- DADA, J. O.; SPASIĆ, I.; PATON, N. W. und MENDES, P. (2010). *SBRML: a markup language for associating systems biology data with models*. In: *Bioinformatics* 26:7, S. 932–938. DOI: [10.1093/bioinformatics/btq069](https://doi.org/10.1093/bioinformatics/btq069).
- DAVIS, R.; SHROBE, H. und SZOLOVITS, P. (1993). *What Is a Knowledge Representation?* In: *AI Magazine* 14:1, S. 17–33. DOI: [10.1609/aimag.v14i1.1029](https://doi.org/10.1609/aimag.v14i1.1029).
- DE JONG, H. (2004). *Qualitative simulation and related approaches for the analysis of dynamic systems*. In: *The Knowledge Engineering Review* 19 (02 Juni 2004), S. 93–132. DOI: [10.1017/S0269888904000177](https://doi.org/10.1017/S0269888904000177).
- DE JONG, H.; GEISELMANN, J.; BATT, G.; HERNANDEZ, C. und PAGE, M. (2004). *Qualitative simulation of the initiation of sporulation in Bacillus subtilis*. In: *Bulletin of Mathematical Biology* 66:2, S. 261–299. DOI: [10.1016/j.bulm.2003.08.009](https://doi.org/10.1016/j.bulm.2003.08.009).
- DE WAARD, A. und SCHNEIDER, J. (2012). *Formalising Uncertainty: An Ontology of Reasoning, Certainty and Attribution (ORCA)*. In: *Proceedings of the Joint Workshop on Semantic Technologies Applied to Biomedical Informatics and Individualized Medicine (SATBI+SWIM)*. Hrsg. von A. R. GONZÁLEZ; J. PATHAK; M. WILKINSON; N. SHAH; R. STEVENS; R. D. BOYCE und Á. G. CRESPO. CEUR Workshop Proceedings 930. Sun SITE Central Europe, RWTH Aachen. URL: <http://ceur-ws.org/Vol-930/p2.pdf>.
- DECRAENE, J. und HINZE, T. (2010). *A Multidisciplinary Survey of Computational Techniques for the Modelling, Simulation and Analysis of Biochemical Networks*. In: *Journal of Universal Computer Science* 16:9 (Mai 2010), S. 1152–1175. URL: [http://www.jucs.org/jucs\\_16/a\\_multidisciplinary\\_survey\\_of](http://www.jucs.org/jucs_16/a_multidisciplinary_survey_of).
- DEGTYARENKO, K.; DE MATOS, P.; ENNIS, M.; HASTINGS, J.; ZBINDEN, M.; MCNAUGHT, A.; ALCÁNTARA, R.; DARSOW, M.; GUEDJ, M. und ASHBURNER, M. (2008). *ChEBI: a database and ontology for chemical entities of biological interest*. In: *Nucleic Acids Research* 36:suppl 1, S. D344–D350. DOI: [10.1093/nar/gkm791](https://doi.org/10.1093/nar/gkm791).

- DEMIR, E.; CARY, M. P.; PALEY, S.; FUKUDA, K.; LEMER, C.; VASTRIK, I.; WU, G.; D'EUSTACHIO, P.; SCHAEFER, C.; LUCIANO, J.; SCHACHERER, F.; MARTINEZ-FLORES, I.; HU, Z.; JIMENEZ-JACINTO, V.; JOSHI-TOPE, G.; KANDASAMY, K.; LOPEZ-FUENTES, A. C.; MI, H.; PICHLER, E.; RODCHENKOV, I.; SPLENDIANI, A.; TKACHEV, S.; ZUCKER, J.; GOPINATH, G.; RAJASIMHA, H.; RAMAKRISHNAN, R.; SHAH, I.; SYED, M.; ANWAR, N.; BABUR, O.; BLINOV, M.; BRAUNER, E.; CORWIN, D.; DONALDSON, S.; GIBBONS, F.; GOLDBERG, R.; HORNBECK, P.; LUNA, A.; MURRAY-RUST, P.; NEUMANN, E.; RUEBENACKER, O.; SAMWALD, M.; VAN IERSEL, M.; WIMALARATNE, S.; ALLEN, K.; BRAUN, B.; WHIRL-CARRILLO, M.; CHEUNG, K.-H.; DAHLQUIST, K.; FINNEY, A.; GILLESPIE, M.; GLASS, E.; GONG, L.; HAW, R.; HONIG, M.; HUBAUT, O.; KANE, D.; KRUPA, S.; KUTMON, M.; LEONARD, J.; MARKS, D.; MERBERG, D.; PETRI, V.; PICO, A.; RAVENSCROFT, D.; REN, L.; SHAH, N.; SUNSHINE, M.; TANG, R.; WHALEY, R.; LETOVKSY, S.; BUETOW, K. H.; RZHETSKY, A.; SCHACHTER, V.; SOBRAL, B. S.; DOGRUSOZ, U.; MCWEENEY, S.; ALADJEM, M.; BIRNEY, E.; COLLADO-VIDES, J.; GOTO, S.; HUCKA, M.; LE NOVÈRE, N.; MALTSEV, N.; PANDEY, A.; THOMAS, P.; WINGENDER, E.; KARP, P. D.; SANDER, C. und BADER, G. D. (2010). *The BioPAX community standard for pathway data sharing*. In: *Nature Biotechnology* 28:9 (Sep. 2010), S. 935–942. DOI: [10.1038/nbt.1666](https://doi.org/10.1038/nbt.1666).
- DITTRICH, P. und DI FENIZIO, P. (2007). *Chemical Organisation Theory*. In: *Bulletin of Mathematical Biology* 69:4, S. 1199–1231. DOI: [10.1007/s11538-006-9130-8](https://doi.org/10.1007/s11538-006-9130-8).
- DOYLE, F. J. und STELLING, J. (2006). *Systems interface biology*. In: *Journal of The Royal Society Interface* 3:10, S. 603–616. DOI: [10.1098/rsif.2006.0143](https://doi.org/10.1098/rsif.2006.0143).
- DRAGER, A.; HASSIS, N.; SUPPER, J.; SCHRODER, A. und ZELL, A. (2008). *SBMLsqueezer: A CellDesigner plug-in to generate kinetic rate equations for biochemical networks*. In: *BMC Systems Biology* 2:1, S. 39. DOI: [10.1186/1752-0509-2-39](https://doi.org/10.1186/1752-0509-2-39).
- DUMONTIER, M. und HOEHNDORF, R. (2010). *Realism for scientific ontologies*. In: *Proceedings of the 2010 conference on Formal Ontology in Information Systems: Proceedings of the Sixth International Conference (FOIS 2010)*. Amsterdam, The Netherlands: IOS Press, S. 387–399. URL: <http://dl.acm.org/citation.cfm?id=1804715.1804755>.
- ELLNER, S. P. und GUCKENHEIMER, J. (2006). *Dynamic Models in Biology*. Princeton, NJ, USA: Princeton University Press. ISBN: 978-0-691-11843-7.
- ERDEN, M. S.; KOMOTO, H.; VAN BEEK, T.; D'AMELIO, V.; ECHAVARRIA, E. und TOMIYAMA, T. (2008). *A review of function modeling: Approaches and applications*. In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 22:02, S. 147–169. DOI: [10.1017/S0890060408000103](https://doi.org/10.1017/S0890060408000103).
- FALKENHAINER, B. und FORBUS, K. D. (1991). *Compositional modeling: finding the right model for the job*. In: *Artificial Intelligence* 51:1-3, S. 95–143.
- FERNÁNDEZ-LÓPEZ, M. und GÓMEZ-PÉREZ, A. (2004). *Searching for a Time Ontology for Semantic Web Applications*. In: *Formal Ontology in Information Systems Proceedings of the Third International Conference (FOIS-2004)*. Hrsg. von A. VARZI und L. VIEU. Frontiers in Artificial Intelligence and Applications 114. IOS Press, Okt. 2004.
- FISHER, J. und HENZINGER, T. A. (2007). *Executable cell biology*. In: *Nature Biotechnology* 25:11 (Nov. 2007), S. 1239–1249. DOI: [10.1038/nbt1356](https://doi.org/10.1038/nbt1356).
- FLEISCHMANN, A.; DARSOW, M.; DEGTYARENKO, K.; FLEISCHMANN, W.; BOYCE, S.; AXELSEN, K. B.; BAIROCH, A.; SCHOMBURG, D.; TIPTON, K. F. und APWEILER, R.



- (2004). *IntEnz, the integrated relational enzyme database*. In: *Nucleic Acids Research* 32:suppl 1 (Jan. 2004): *Database issue*, S. 434–437. DOI: [10.1093/nar/gkh119](https://doi.org/10.1093/nar/gkh119).
- FORBUS, K. D. (1984). *Qualitative process theory*. In: *Artificial Intelligence* 24:1-3, S. 85–168. DOI: [10.1016/0004-3702\(84\)90038-9](https://doi.org/10.1016/0004-3702(84)90038-9).
- FORBUS, K. D. (2008). *Qualitative Modeling*. In: Hrsg. von F. VAN HARMELEN; V. LIFSCHITZ und B. PORTER. 1. Aufl. *Foundations of Artificial Intelligence* 3. Elsevier Science. Kap. 9, S. 361–393. DOI: [10.1016/S1574-6526\(07\)03009-X](https://doi.org/10.1016/S1574-6526(07)03009-X).
- FUNAHASHI, A.; MOROHASHI, M.; KITANO, H. und TANIMURA, N. (2003). *CellDesigner: a process diagram editor for gene-regulatory and biochemical networks*. In: *BIOSILICO* 1:5, S. 159–162. DOI: [10.1016/S1478-5382\(03\)02370-9](https://doi.org/10.1016/S1478-5382(03)02370-9).
- GABRIEL, G. (2010). *Logische Präzision und ästhetische Prägnanz. Abschiedsvorlesung von Prof. Dr. Gottfried Gabriel an der FSU Jena am 21. April 2009*. In: *Literaturwissenschaftliches Jahrbuch* 51. Hrsg. von V. KAPP; K. MÜLLER; K. RIDDER; R. WIMMER und J. ZIMMERMANN, S. 375–390.
- GÄRDENFORS, P. (2004). *How to Make the Semantic Web More Semantic*. In: *Formal Ontology in Information Systems Proceedings of the Third International Conference (FOIS-2004)*. Hrsg. von A. VARZI und L. VIEU. *Frontiers in Artificial Intelligence and Applications* 114. IOS Press, Okt. 2004, S. 17–34.
- GENE ONTOLOGY CONSORTIUM (2001). *Creating the gene ontology resource: design and implementation*. In: *Genome Research* 11:8 (Aug. 2001), S. 1425–1433. DOI: [10.1101/gr.180801](https://doi.org/10.1101/gr.180801).
- GENESERITH, M. und FIKES, R. (1992). *Knowledge interchange format, version 3.0. Reference manual*. Manual Logic 92-1. Stanford University. URL: <http://www-ksl.stanford.edu/knowledge-sharing/papers/kif.ps>.
- GENNARI, J. H.; NEAL, M. L.; CARLSON, B. E. und COOK, D. L. (2008). *Integration of multi-scale biosimulation models via light-weight semantics*. In: *Pacific Symposium on Biocomputing*, S. 414–425. URL: <http://www.ncbi.nlm.nih.gov/pubmed/18229704>.
- GERO, J. S. (1990). *Design prototypes: a knowledge representation schema for design*. In: *AI Magazine* 11:4, S. 26–36. URL: <http://portal.acm.org/citation.cfm?id=95793>.
- GKOUTOS, G. V.; SCHOFIELD, P. N. und HOEHNDORF, R. (2012). *The Units Ontology: a tool for integrating units of measurement in science*. In: *Database* 2012. DOI: [10.1093/database/bas033](https://doi.org/10.1093/database/bas033).
- GKOUTOS, G.; GREEN, E.; MALLON, A.-M.; HANCOCK, J. und DAVIDSON, D. (2004). *Using ontologies to describe mouse phenotypes*. In: *Genome Biology* 6:1, R8. DOI: [10.1186/gb-2004-6-1-r8](https://doi.org/10.1186/gb-2004-6-1-r8).
- GOEL, A. K.; RUGABER, S. und VATTAM, S. (2009). *Structure, behavior, and function of complex systems: The structure, behavior, and function modeling language*. In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 23:1, S. 23–35. DOI: [10.1017/S0890060409000080](https://doi.org/10.1017/S0890060409000080).
- GOLDBETER, A. (1991). *A minimal cascade model for the mitotic oscillator involving cyclin and cdc2 kinase*. In: *Proceedings of the National Academy of Sciences of the United States of America* 88:20 (Okt. 1991), S. 9107–9111. DOI: [10.1073/pnas.88.20.9107](https://doi.org/10.1073/pnas.88.20.9107).
- GÖRZ, G.; ROLLINGER, C.-R. und SCHNEEBERGER, J., Hrsg. (2000). *Handbuch der Künstlichen Intelligenz*. 3. Aufl. München, Wien: Oldenbourg Verlag. ISBN: 978-3-486-25049-7.

- GRENON, P.; SMITH, B. und GOLDBERG, L. (2004). *Biodynamic ontology: applying BFO in the biomedical domain*. In: *Studies in health technology and informatics* 102, S. 20–38. URL: <http://www.ncbi.nlm.nih.gov/pubmed/15853262>.
- GRUBER, T. R. (1993a). *A translation approach to portable ontology specifications*. In: *Knowledge Acquisition* 5:2, S. 199–220. DOI: [10.1006/knac.1993.1008](https://doi.org/10.1006/knac.1993.1008).
- GRUBER, T. R. (1993b). *Towards Principles for the Design of Ontologies Used for Knowledge Sharing*. In: *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Hrsg. von N. GUARINO und R. POLI. Deventer, The Netherlands: Kluwer Academic Publishers.
- GRUBER, T. R. und OLSEN, G. R. (1994). *An Ontology for Engineering Mathematics*. In: *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*. Hrsg. von J. DOYLE; E. SANDEWALL und P. TORASSO. Morgan Kaufmann, S. 258–269. URL: <http://ksl.stanford.edu/knowledge-sharing/papers/engmath.html>.
- GUARINO, N. (1998). *Formal Ontology and Information Systems*. In: *Formal Ontology in Information Systems – Proceedings of FOIS 1998*. Hrsg. von N. GUARINO. Frontiers in Artificial Intelligence and Applications 46. Amsterdam, The Netherlands: IOS Press, S. 3–15.
- GUARINO, N.; CARRARA, M. und GIARETTA, P. (1994). *Formalizing Ontological Commitment*. In: *Proceedings of the twelfth national conference on artificial intelligence*. AAAI Press, S. 560–567.
- GUARINO, N. und GIARETTA, P. (1995). *Ontologies and Knowledge Bases: Towards a Terminological Clarification*. In: *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*. Hrsg. von N. MARS. IOS Press, S. 25–32.
- GUCKENHEIMER, J. und HOLMES, P. (1983). *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields*. Applied mathematical sciences 42. New York: Springer. ISBN: 0-387-90819-6.
- GUHA, R. und BRICKLEY, D. (2004). *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation. World Wide Web Consortium. 10. Feb. 2004. URL: <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>.
- HALE, J. K. und KOÇAK, H. (1991). *Dynamics and Bifurcations*. Texts in Applied Mathematics. New York: Springer-Verlag. ISBN: 978-3-540-97141-2.
- HARNAD, S. (1990). *The symbol grounding problem*. In: *Physica D: Nonlinear Phenomena* 42:1–3, S. 335–346. DOI: [10.1016/0167-2789\(90\)90087-6](https://doi.org/10.1016/0167-2789(90)90087-6).
- HARNAD, S. (1994). *Levels of functional equivalence in reverse bioengineering*. In: *Artificial Life* 1:3, S. 293–301. DOI: [10.1162/artl.1994.1.293](https://doi.org/10.1162/artl.1994.1.293).
- HARRIS, S. und SEABORNE, A., Hrsg. (2013). *SPARQL 1.1 Query Language*. W3C Recommendation. World Wide Web Consortium. 21. März 2013. URL: <http://www.w3.org/TR/2013/REC-sparql11-query-20130321/>.
- HAYES, P. J. (1977). *In defense of logic*. In: *Proceedings of the 5th international joint conference on Artificial intelligence - Volume 1 (IJCAI-77)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., S. 559–565. URL: <http://dl.acm.org/citation.cfm?id=1624435.1624559>.
- HAYES, P. (1978). *Naive Physics I: Ontology for Liquids*. Working Paper 35. Geneva: Institute for Semantic und Cognitive Studies. URL: <http://www.issco.unige.ch/working-papers/Hayes-1978-35.pdf>.



- HAYES, P. (1979). *The Naive Physics Manifesto*. In: *Expert Systems in the Micro-Electronic Age*. Hrsg. von D. MICHIE. Edinburgh: Edinburgh University Press, S. 242–270.
- HAYES, P. (1985a). *Naive Physics I: Ontology for Liquids*. In: *Formal Theories of the Commonsense World*. Hrsg. von J. R. HOBBS und R. C. MOORE. Reprint of HAYES 1978. Norwood, NJ: Ablex Publishing Company. Kap. 3, S. 71–107.
- HAYES, P. (1985b). *The Second Naive Physics Manifest*. In: *Formal Theories of the Common Sense World*. Hrsg. von J. R. HOBBS und R. C. MOORE. Norwood, NJ: Ablex Publishing Company. Kap. 1, S. 1–36.
- HEMPEL, C. G. und LENZEN, W. (1977). *Aspekte wissenschaftlicher Erklärung*. Grundlagen der Kommunikation und Kognition / Foundations of Communication and Cognition. De Gruyter. ISBN: 978-3-11-083048-4.
- HINDMARSH, A. C. (1983). *ODEPACK, A Systematized Collection of ODE Solvers*. In: *Scientific computing: applications of mathematics and computing to the physical sciences*. 10th IMACS World Congress on Systems Simulation and Scientific Computation. Hrsg. von R. STEPLEMAN; M. CARVER; R. PESKIN; W. AMES und R. VICHNEVETSKY. Bd. 1. IMACS Transactions on Scientific Computation. Amsterdam: North-Holland Publishing Company, S. 55–64. URL: <http://computation.llnl.gov/casc/nsde/pubs/u88007.pdf>.
- HINDMARSH, A. C.; BROWN, P. N.; GRANT, K. E.; LEE, S. L.; SERBAN, R.; SHUMAKER, D. E. und WOODWARD, C. S. (2005). *SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers*. In: *ACM Transactions on Mathematical Software* 31:3 (Sep. 2005), S. 363–396. DOI: [10.1145/1089014.1089020](https://doi.org/10.1145/1089014.1089020).
- HITZLER, P.; KRÖTZSCH, M.; PARSIA, B.; PATEL-SCHNEIDER, P. F. und RUDOLPH, S., Hrsg. (2012). *OWL 2 Web Ontology Language Primer (Second Edition)*. W3C Recommendation. World Wide Web Consortium. 11. Dez. 2012. URL: <http://www.w3.org/TR/2012/REC-owl2-primer-20121211/>.
- HOFMANN, M. (2013). *Ontologies in Modeling and Simulation: An Epistemological Perspective*. In: *Ontology, Epistemology, and Teleology for Modeling and Simulation: Philosophical Foundations for Intelligent M&S Applications*. Hrsg. von A. TOLK. Bd. 44. Intelligent Systems Reference Library. Berlin: Springer, S. 59–87. DOI: [10.1007/978-3-642-31140-6](https://doi.org/10.1007/978-3-642-31140-6).
- HOOPS, S.; SAHLE, S.; GAUGES, R.; LEE, C.; PAHLE, J.; SIMUS, N.; SINGHAL, M.; XU, L.; MENDES, P. und KUMMER, U. (2006). *COPASI – a COMplex PATHway SIMulator*. In: *Bioinformatics* 22:24, S. 3067–3074. DOI: [10.1093/bioinformatics/btl485](https://doi.org/10.1093/bioinformatics/btl485).
- HORRIDGE, M. und PATEL-SCHNEIDER, P. F. (2012). *OWL 2 Web Ontology Language: Manchester Syntax*. 2. Aufl. W3C Working Group Note. World Wide Web Consortium. 11. Dez. 2012. URL: <http://www.w3.org/TR/2012/NOTE-owl2-manchester-syntax-20121211/>.
- HORROCKS, I.; PATEL-SCHNEIDER, P. F.; BOLEY, H.; TABET, S.; GROSOFF, B. und DEAN, M. (2004). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. W3C Member Submission. World Wide Web Consortium. 21. Mai 2004. URL: <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>.
- HUBBARD, J. und WEST, B. (1995). *Differential Equations: A Dynamical Systems Approach. Higher Dimensional Systems*. Applications of Mathematics 18. Springer. ISBN: 978-0-387-94377-0.
- HUCKA, M.; FINNEY, A.; SAURO, H. M.; BOLOURI, H.; DOYLE, J. und KITANO, H. (2002). *The ERATO Systems Biology Workbench: enabling interaction and exchange*

- between software tools for computational biology. In: *Pacific Symposium on Biocomputing*, S. 450–461. URL: <http://www.ncbi.nlm.nih.gov/pubmed/11928498>.
- HUCKA, M.; FINNEY, A.; SAURO, H. M.; BOLOURI, H.; DOYLE, J. C.; KITANO, H.; ARKIN, A. P.; BORNSTEIN, B. J.; BRAY, D.; CORNISH-BOWDEN, A.; CUELLAR, A. A.; DRONOV, S.; GILLES, E. D.; GINKEL, M.; GOR, V.; GORYANIN, I. I.; HEDLEY, W. J.; HODGMAN, T. C.; HOFMEYR, J.-H. S.; HUNTER, P. J.; JUTY, N. S.; KASBERGER, J. L.; KREMLING, A.; KUMMER, U.; LE NOVÈRE, N.; LOEW, L. M.; LUCIO, D.; MENDES, P.; MINCH, E.; MJOLSNES, E. D.; NAKAYAMA, Y.; NELSON, M. R.; NIELSEN, P. F.; SAKURADA, T.; SCHAFF, J. C.; SHAPIRO, B. E.; SHIMIZU, T. S.; SPENCE, H. D.; STELLING, J.; TAKAHASHI, K.; TOMITA, M.; WAGNER, J. M. und WANG, J. (2003). *The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models*. In: *Bioinformatics* 19:4 (März 2003), S. 524–531.
- HUCKA, M.; HOOPS, S.; KEATING, S. M.; LE NOVÈRE, N.; SAHLE, S. und WILKINSO, D. J. (2008). *Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions*. Language Specification. Version 4, Release 1. COMBINE. 22. Dez. 2008. URL: <http://identifiers.org/combine.specifications/sbml.level-2.version-4.release-1>.
- IDEKER, T.; GALITSKI, T. und HOOD, L. (2001). *A new approach to decoding life: systems biology*. In: *Annual Review of Genomics and Human Genetics* 2, S. 343–372. DOI: [10.1146/annurev.genom.2.1.343](https://doi.org/10.1146/annurev.genom.2.1.343).
- ISO (2004). *ISO 8601:2004. Data elements and interchange formats – Information interchange – Representation of dates and times*. Norm. International Organization for Standardization. Geneva, Switzerland. URL: <http://www.iso.org/iso/iso8601>.
- IWASAKI, Y. und LOW, C. M. (1991). *Model Generation and Simulation of Device Behavior with Continuous and Discrete Changes*. Techn. Ber. KSL-91-69. Stanford University, Knowledge Systems Lab.
- IZHIKEVICH, E. M. (2011). *Encyclopedia of dynamical systems*. In: *Scholarpedia* 6:11, S. 1287. DOI: [10.4249/scholarpedia.1287](https://doi.org/10.4249/scholarpedia.1287).
- IZHIKEVICH, E. M. (2007). *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. Computational Neuroscience. MIT Press. ISBN: 978-0-262-09043-8.
- JACKSON, E. A. (1989). *Perspectives of Nonlinear Dynamics*. Bd. 1. Cambridge: Cambridge University Press. ISBN: 978-0-521-34504-0.
- JANSEN, L. und SMITH, B., Hrsg. (2008). *Biomedizinische Ontologie: Wissen strukturieren für den Informatik-Einsatz*. vdf Hochschulverlag AG an der ETH Zürich. ISBN: 978-3-7281-3183-6.
- JETSCHKE, G. (1989). *Mathematik der Selbstorganisation*. Berlin: VEB Deutscher Verlag der Wissenschaften. ISBN: 978-3-326-00384-9.
- JONES, A. R.; MILLER, M.; AEBERSOLD, R.; APWEILER, R.; BALL, C. A.; BRAZMA, A.; DEGREEF, J.; HARDY, N.; HERMJAKOB, H.; HUBBARD, S. J.; HUSSEY, P.; IGRA, M.; JENKINS, H.; JULIAN, R. K.; LAURSEN, K.; OLIVER, S. G.; PATON, N. W.; SANSONE, S.-A.; SARKANS, U.; STOECKERT, C. J.; TAYLOR, C. F.; WHETZEL, P. L.; WHITE, J. A.; SPELLMAN, P. und PIZARRO, A. (2007). *The Functional Genomics Experiment model (FuGE): an extensible framework for standards in functional genomics*. In: *Nature Biotechnology* 25:10 (Okt. 2007), S. 1127–1133. DOI: [10.1038/nbt1347](https://doi.org/10.1038/nbt1347).
- JOSHI-TOPE, G.; GILLESPIE, M.; VASTRIK, I.; D'EUSTACHIO, P.; SCHMIDT, E.; DE BONO, B.; JASSAL, B.; GOPINATH, G.; WU, G.; MATTHEWS, L.; LEWIS, S.; BIRNEY, E. und

- STEIN, L. (2005). *Reactome: a knowledgebase of biological pathways*. In: *Nucleic Acids Research* 33:suppl 1, S. D428–432. DOI: [10.1093/nar/gki072](https://doi.org/10.1093/nar/gki072).
- JUNKER, B.; KLUKAS, C. und SCHREIBER, F. (2006). *VANTED: A system for advanced data analysis and visualization in the context of biological networks*. In: *BMC Bioinformatics* 7:1, S. 109. DOI: [10.1186/1471-2105-7-109](https://doi.org/10.1186/1471-2105-7-109).
- JUTY, N.; LE NOVÈRE, N. und LAIBE, C. (2012). *Identifiers.org and MIRIAM Registry: community resources to provide persistent identification*. In: *Nucleic Acids Research* 40:D1 (Jan. 2012): *Database issue*, S. 580–586. DOI: [10.1093/nar/gkr1097](https://doi.org/10.1093/nar/gkr1097).
- KELL, D. B. und OLIVER, S. G. (2004). *Here is the evidence, now what is the hypothesis? The complementary roles of inductive and hypothesis-driven science in the post-genomic era*. In: *Bioessays* 26:1 (Jan. 2004), S. 99–105. DOI: [10.1002/bies.10385](https://doi.org/10.1002/bies.10385).
- KESELER, I. M.; MACKIE, A.; PERALTA-GIL, M.; SANTOS-ZAVALA, A.; GAMA-CASTRO, S.; BONAVIDES-MARTÍNEZ, C.; FULCHER, C.; HUERTA, A. M.; KOTHARI, A.; KRUMMENACKER, M.; LATENDRESSE, M.; MUÑIZ-RASCADO, L.; ONG, Q.; PALEY, S.; SCHRÖDER, I.; SHEARER, A. G.; SUBHRAVETI, P.; TRAVERS, M.; WEERASINGHE, D.; WEISS, V.; COLLADO-VIDES, J.; GUNSALUS, R. P.; PAULSEN, I. und KARP, P. D. (2013). *EcoCyc: fusing model organism databases with systems biology*. In: *Nucleic Acids Research* 41:D1: *Database issue*, S. D605–D612. DOI: [10.1093/nar/gks1027](https://doi.org/10.1093/nar/gks1027).
- KING, R. D.; GARRETT, S. M. und COGHILL, G. M. (2005). *On the use of qualitative reasoning to simulate and identify metabolic pathways*. In: *Bioinformatics* 21:9, S. 2017–2026. DOI: [10.1093/bioinformatics/bti255](https://doi.org/10.1093/bioinformatics/bti255).
- KITANO, H. (2002a). *Computational systems biology*. In: *Nature* 420:6912 (Nov. 2002), S. 206–210. DOI: [10.1038/nature01254](https://doi.org/10.1038/nature01254).
- KITANO, H. (2002b). *Standards for modeling*. In: *Nature Biotechnology* 20:4 (Apr. 2002), S. 337. DOI: [10.1038/nbt0402-337a](https://doi.org/10.1038/nbt0402-337a).
- KLEER, J. D. und BROWN, J. S. (1984). *A qualitative physics based on confluences*. In: *Artificial Intelligence* 24:1–3, S. 7–83. DOI: [10.1016/0004-3702\(84\)90037-7](https://doi.org/10.1016/0004-3702(84)90037-7).
- KLIPP, E.; LIEBERMEISTER, W.; WIERLING, C.; KOWALD, A.; LEHRACH, H. und HERWIG, R. (2009). *Systems Biology: A Textbook*. Weinheim: Wiley-VCH. ISBN: 978-3-527-31874-2.
- KLIR, G. J. (1985). *Architecture of Systems Problem Solving*. New York: Plenum Press. ISBN: 978-0-306-41867-9.
- KNÜPFER, C. (2003). *Testdatengenerierung für die Inferenz von Genregulationsnetzen*. Diplomarbeit. Jena: Friedrich-Schiller-Universität.
- KNÜPFER, C. und BECKSTEIN, C. (2013a). *Function of Dynamic Models in Systems Biology: Linking Structure to Behaviour*. In: *Journal of Biomedical Semantics* 4:1, S. 24. DOI: [10.1186/2041-1480-4-24](https://doi.org/10.1186/2041-1480-4-24).
- KNÜPFER, C.; BECKSTEIN, C. und DITTRICH, P. (2006). *Towards a Semantic Description of Bio-Models: Meaning Facets – A Case Study*. In: *Proceedings of the Second International Symposium on Semantic Mining in Biomedicine (SMBM 2006)*. Hrsg. von S. ANANIADOU und J. FLUCK. CEUR Workshop Proceedings 177. Sun SITE Central Europe, RWTH Aachen, S. 97–100. URL: [http://ceur-ws.org/Vol-177/poster\\_knuepfer.pdf](http://ceur-ws.org/Vol-177/poster_knuepfer.pdf).
- KNÜPFER, C.; BECKSTEIN, C.; DITTRICH, P. und LE NOVÈRE, N. (2013b). *Structure, Function, and Behaviour of Computational Models in Systems Biology*. In: *BMC Systems Biology* 7:1, S. 43. DOI: [10.1186/1752-0509-7-43](https://doi.org/10.1186/1752-0509-7-43).
- KOOLMAN, J.; RÖHM, K.-H. und WIRTH, J. (2003). *Taschenatlas der Biochemie*. Stuttgart [u.a.]: Thieme. ISBN: 978-3-13-759403-1.

- KRAUSE, F.; UHLENDORF, J.; LUBITZ, T.; SCHULZ, M.; KLIPP, E. und LIEBERMEISTER, W. (2010). *Annotation and merging of SBML models with semanticSBML*. In: *Bioinformatics* 26:3, S. 421–422. DOI: [10.1093/bioinformatics/btp642](https://doi.org/10.1093/bioinformatics/btp642).
- KRIETE, A. und EILS, R. (2006). *Computational Systems Biology*. Elsevier Academic Press. ISBN: 978-0-12-088786-6.
- KRIPKE, S. A. (1963). *Semantical Analysis of Modal Logic I: Normal Modal Propositional Calculi*. In: *Mathematical Logic Quarterly* 9:5-6, S. 67–96. DOI: [10.1002/malq.19630090502](https://doi.org/10.1002/malq.19630090502).
- KROHS, U. und KROES, P. (2009). *Philosophical Perspectives on Organismic and Artifactual Functions*. In: *Functions in Biological and Artificial Worlds: Comparative Philosophical Perspectives*. Hrsg. von U. KROHS und P. KROES. Vienna Series in Theoretical Biology. MIT Press.
- KUIPERS, B. J. (1986). *Qualitative simulation*. In: *Artificial Intelligence* 29:3, S. 289–338. DOI: [10.1016/0004-3702\(86\)90073-1](https://doi.org/10.1016/0004-3702(86)90073-1).
- KUIPERS, B. J. (1994). *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. Cambridge, MA: MIT Press. ISBN: 978-0-262-11190-4.
- KUMAR, A. und SMITH, B. (2004). *Controlled vocabularies in bioinformatics: a case study in the gene ontology*. In: *Drug Discovery Today: BIOSILICO* 2:6 (Nov. 2004), S. 246–252. DOI: [10.1016/S1741-8364\(04\)02424-2](https://doi.org/10.1016/S1741-8364(04)02424-2).
- KUNZE, J. und BAKER, T. (2007). *The Dublin Core Metadata Element Set*. RFC 5013. Internet Engineering Task Force, Aug. 2007. URL: <http://www.ietf.org/rfc/rfc5013.txt>.
- KUZNETSOV, Y. A. (1998). *Elements of applied bifurcation theory*. 2. Aufl. New York, NY, USA: Springer-Verlag. ISBN: 978-0-387-98382-0.
- LANDER, A. D. (2004). *A Calculus of Purpose*. In: *PLOS Biology* 2:6 (Juni 2004), e164. DOI: [10.1371/journal.pbio.0020164](https://doi.org/10.1371/journal.pbio.0020164).
- LE NOVÈRE, N.; BORNSTEIN, B.; BROICHER, A.; COURTOT, M.; DONIZELLI, M.; DHARURI, H.; LI, L.; SAURO, H.; SCHILSTRA, M.; SHAPIRO, B.; SNOEP, J. und HUCKA, M. (2006). *BioModels Database: A Free, Centralized Database of Curated, Published, Quantitative Kinetic Models of Biochemical and Cellular Systems*. In: *Nucleic Acids Research* 34:D1 (Jan. 2006): *Database issue*, S. 689–691.
- LE NOVÈRE, N. (2006a). *Model storage, exchange and integration*. In: *BMC Neuroscience* 7:Suppl 1, S11. DOI: [10.1186/1471-2202-7-S1-S11](https://doi.org/10.1186/1471-2202-7-S1-S11).
- LE NOVÈRE, N. (2006b). *Tyson (1991), Modelling cell division*. Model of the Month. Original model: BIOMD0000000005. BioModels Database, Okt. 2006. URL: <http://www.ebi.ac.uk/biomodels-main/static-pages.do?page=ModelMonth/2006-10>.
- LE NOVÈRE, N. (2010). *Vision of standards interoperability*. Talk at the 2010 SBML-BioModels.net Hackathon. 1. Mai 2010. URL: <http://sbml.org/images/a/a6/Lenovere-intro-2010-05-01.pdf>.
- LE NOVÈRE, N.; FINNEY, A.; HUCKA, M.; BHALLA, U. S.; CAMPAGNE, F.; COLLADO-VIDES, J.; CRAMPIN, E. J.; HALSTEAD, M.; KLIPP, E.; MENDES, P.; NIELSEN, P.; SAURO, H.; SHAPIRO, B.; SNOEP, J. L.; SPENCE, H. D. und WANNER, B. L. (2005). *Minimum information requested in the annotation of biochemical models (MIRIAM)*. In: *Nature Biotechnology* 23:12 (Dez. 2005), S. 1509–1515. DOI: [10.1038/nbt1156](https://doi.org/10.1038/nbt1156).
- LE NOVÈRE, N.; HUCKA, M.; MI, H.; MOODIE, S.; SCHREIBER, F.; SOROKIN, A.; DEMIR, E.; WEGNER, K.; ALADJEM, M. I.; WIMALARATNE, S. M.; BERGMAN, F. T.; GAUGES,

- R.; GHAZAL, P.; KAWAJI, H.; LI, L.; MATSUOKA, Y.; VILLÉGER, A.; BOYD, S. E.; CALZONE, L.; COURTOT, M.; DOGRUSOZ, U.; FREEMAN, T. C.; FUNAHASHI, A.; GHOSH, S.; JOURAKU, A.; KIM, S.; KOLPAKOV, F.; LUNA, A.; SAHLE, S.; SCHMIDT, E.; WATTERSON, S.; WU, G.; GORYANIN, I.; KELL, D. B.; SANDER, C.; SAURO, H.; SNOEP, J. L.; KOHN, K. und KITANO, H. (2009). *The Systems Biology Graphical Notation*. In: *Nature Biotechnology* 27:8 (Aug. 2009), S. 735–741. DOI: [10.1038/nbt.1558](https://doi.org/10.1038/nbt.1558).
- LE NOVÈRE, N. und KNÜPFER, C. (2007a). *TEDDY – A Terminology for the Description of the Dynamics of Bio-Models*. In: *Foundations of Systems Biology in Engineering (FOSBE)*. URL: [http://www.ebi.ac.uk/~lenov/POSTERS/Poster\\_TEDDY\\_FOSBE2007.pdf](http://www.ebi.ac.uk/~lenov/POSTERS/Poster_TEDDY_FOSBE2007.pdf).
- LE NOVÈRE, N.; LAIBE, C.; COURTOT, M.; KNÜPFER, C. und HUCKA, M. (2007b). *Semantic Annotation of Kinetic Models in Systems Biology*. In: *The Eighth International Conference on Systems Biology (ICSB)*, G28. URL: <http://www.icsb07.caltech.edu/proceedings/abstracts/G28.pdf>.
- LE NOVÈRE, N.; LAIBE, C.; COURTOT, M.; KNÜPFER, C. und HUCKA, M. (2007c). *Semantic Annotation of Kinetic Models in Systems Biology*. In: *15th Annual International Conference on Intelligent Systems for Molecular Biology*, S. C126. URL: [https://www.ebi.ac.uk/compneur-srv/miriam/elements/Poster\\_Semantics\\_ISMB-2007.pdf](https://www.ebi.ac.uk/compneur-srv/miriam/elements/Poster_Semantics_ISMB-2007.pdf).
- LEE, J. A.; SPIDLEN, J.; BOYCE, K.; CAI, J.; CROSBIE, N.; DALPHIN, M.; FURLONG, J.; GASPARETTO, M.; GOLDBERG, M.; GORALCZYK, E. M.; HYUN, B.; JANSEN, K.; KOLLMANN, T.; KONG, M.; LEIF, R.; MCWEENEY, S.; MOLOSHOK, T. D.; MOORE, W.; NOLAN, G.; NOLAN, J.; NIKOLICH-ZUGICH, J.; PARRISH, D.; PURCELL, B.; QIAN, Y.; SELVARAJ, B.; SMITH, C.; TCHUVATKINA, O.; WERTHEIMER, A.; WILKINSON, P.; WILSON, C.; WOOD, J.; ZIGON, R.; SCHEUERMANN, R. H. und BRINKMAN, R. R. (2008). *MIFlowCyt: The minimum information about a flow cytometry experiment*. In: *Cytometry Part A* 73A:10, S. 926–930. DOI: [10.1002/cyto.a.20623](https://doi.org/10.1002/cyto.a.20623).
- LEIFHEIT, J. und KING, R. (2005). *Systematic structure and parameter identification for biological systems supported by a software-tool*. In: *Proceedings of 16th IFAC World Congress*. Hrsg. von P. PIZTEK. Elsevier.
- LENAT, D. B. und GUHA, R. V. (1990). *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project*. Reading, MA: Addison-Wesley. ISBN: 978-0-201-51752-1.
- LISTER, A. L.; POCKOCK, M.; TASCHUK, M. und WIPAT, A. (2009). *Saint: a lightweight integration environment for model annotation*. In: *Bioinformatics* 25:22, S. 3026–3027. DOI: [10.1093/bioinformatics/btp523](https://doi.org/10.1093/bioinformatics/btp523).
- LLOYD, C. M.; LAWSON, J. R.; HUNTER, P. J. und NIELSEN, P. F. (2008). *The CellML Model Repository*. In: *Bioinformatics* 24:18, S. 2122–2123. DOI: [10.1093/bioinformatics/btn390](https://doi.org/10.1093/bioinformatics/btn390).
- MACHAMER, P.; DARDEN, L. und CRAVER, C. F. (2000). *Thinking about Mechanisms*. In: *Philosophy of Science* 67:1, S. 1–25. URL: <http://www.jstor.org/stable/188611>.
- MAHR, B. E. (2009). *Die Informatik und die Logik der Modelle*. In: *Informatik-Spektrum* 32:3 (Juni 2009), S. 228–249. DOI: [10.1007/s00287-009-0340-y](https://doi.org/10.1007/s00287-009-0340-y).
- MANDEL, J.; PALFREYMAN, N. M.; LOPEZ, J. A. und DUBITZKY, W. (2004). *Representing bioinformatics causality*. In: *Briefings in Bioinformatics* 5:3 (Sep. 2004), S. 270–283. DOI: [10.1093/bib/5.3.270](https://doi.org/10.1093/bib/5.3.270).



- MANOLA, F. und MILLER, E., Hrsg. (2004). *RDF Primer*. W3C Recommendation. World Wide Web Consortium. 10. Feb. 2004. URL: <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>.
- MCCARTHY, J. (1980). *Circumscription – A Form of Non-Monotonic Reasoning*. In: *Artificial Intelligence* 13, S. 27–39.
- MICHAEL, J.; MEJINO, J. und ROSSE, C. (2001). *The role of definitions in biomedical concept representation*. In: *AMIA Annual Symposium Proceedings*, S. 463–467. URL: <http://www.ncbi.nlm.nih.gov/pubmed/11825231>.
- MILES, A. und BECHHOFFER, S. (2009). *SKOS Simple Knowledge Organization System Reference*. W3C Recommendation. World Wide Web Consortium. 18. Aug. 2009. URL: <http://www.w3.org/TR/2009/REC-skos-reference-20090818/>.
- MILLER, A. K.; YU, T.; BRITTEN, R.; COOLING, M. T.; LAWSON, J.; COWAN, D.; GARNY, A.; HALSTEAD, M. D.; HUNTER, P. J.; NICKERSON, D. P.; NUNNS, G.; WIMALARATNE, S. M. und NIELSEN, P. M. (2011). *Revision history aware repositories of computational models of biological systems*. In: *BMC Bioinformatics* 12, S. 22–22. DOI: [10.1186/1471-2105-12-22](https://doi.org/10.1186/1471-2105-12-22).
- MONTEIRO, P. T.; ROPERS, D.; MATEESCU, R.; FREITAS, A. T. und DE JONG, H. (2008). *Temporal logic patterns for querying dynamic models of cellular interaction networks*. In: *Bioinformatics* 24:16 (Aug. 2008), S. i227–i233. DOI: [10.1093/bioinformatics/btn275](https://doi.org/10.1093/bioinformatics/btn275).
- MOODIE, S.; LE NOVÈRE, N.; DEMIR, E.; MI, H. und SCHREIBER, F. (2009). *Systems biology graphical notation: process description language level 1*. In: *Nature Precedings*. DOI: [10.1038/npre.2009.3721.1](https://doi.org/10.1038/npre.2009.3721.1).
- MORRISON, M. und MORGAN, M. S. (1999). *Models as mediating instruments*. In: *Models as Mediators*. Hrsg. von M. S. MORGAN und M. MORRISON. Ideas in Context 52. Cambridge University Press. Kap. 2, S. 10–37. DOI: [10.1017/CB09780511660108.003](https://doi.org/10.1017/CB09780511660108.003).
- MOTIK, B.; PATEL-SCHNEIDER, P. und PARSIA, B. (2012). *OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (Second Edition)*. W3C Recommendation. World Wide Web Consortium. 11. Dez. 2012. URL: <http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/>.
- MOTIK, B.; SHEARER, R. und HORROCKS, I. (2009). *Hypertableau Reasoning for Description Logics*. In: *Journal of Artificial Intelligence Research* 36, S. 165–228.
- MOULINES, C. U. (1994). *Wer bestimmt, was es gibt? Zum Verhältnis zwischen Ontologie und Wissenschaftstheorie*. In: *Zeitschrift für philosophische Forschung* 48:2, S. 175–191. URL: <http://www.jstor.org/stable/20483610>.
- NARDI, D. und BRACHMAN, R. J. (2003). *An Introduction to Description Logics*. In: *The Description Logic Handbook - Theory, Implementation and Applications*. Hrsg. von F. BAADER; D. CALVANESE; D. MCGUINNESS; D. NARDI und P. PATEL-SCHNEIDER. New York, NY, USA: Cambridge University Press. Kap. 1, S. 5–44.
- NEAL, M. L.; GENNARI, J. H.; ARTS, T. und COOK, D. L. (2009). *Advances in semantic representation for multiscale biosimulation: a case study in merging models*. In: *Pacific Symposium on Biocomputing*, S. 304–315. URL: <http://www.ncbi.nlm.nih.gov/pubmed/19209710>.
- NEBEL, B. (1990). *Reasoning and Revision in Hybrid Representation Systems*. Lecture Notes in Artificial Intelligence 422. Berlin, Heidelberg, New York: Springer-Verlag. ISBN: 978-3-540-52443-4.

- NECHES, R.; FIKES, R.; FININ, T.; GRUBER, T.; PATIL, R.; SENATOR, T. und SWARTOUT, W. (1991). *Enabling Technology for Knowledge Sharing*. In: *AI Magazine* 12:3 (Aug. 1991), S. 36–56. DOI: [10.1609/aimag.v12i3.902](https://doi.org/10.1609/aimag.v12i3.902).
- NEWELL, A. und SIMON, H. A. (1976). *Computer science as empirical inquiry: symbols and search*. In: *Communications of the ACM* 19:3 (März 1976), S. 113–126. DOI: [10.1145/1283920.1283930](https://doi.org/10.1145/1283920.1283930).
- NEWPORT, J. und KIRSCHNER, M. (1982). *A major developmental transition in early xenopus embryos: I. characterization and timing of cellular changes at the midblastula stage*. In: *Cell* 30:3, S. 675–686. DOI: [10.1016/0092-8674\(82\)90272-0](https://doi.org/10.1016/0092-8674(82)90272-0).
- NIC, M.; JIRAT, J. und KOSATA, B. *IUPAC Compendium of Chemical Terminology (Gold Book), Online Version*. DOI: [10.1351/goldbook](https://doi.org/10.1351/goldbook).
- NICOLA, R. und VAANDRAGER, F. (1990). *Action versus state based logics for transition systems*. In: *Semantics of Systems of Concurrent Processes*. Hrsg. von I. GUESSARIAN. Lecture Notes in Computer Science 469. Springer Berlin Heidelberg, S. 407–419. DOI: [10.1007/3-540-53479-2\\_17](https://doi.org/10.1007/3-540-53479-2_17).
- NICOLIS, G. und PRIGOGINE, I. (1977). *Self-Organization in Nonequilibrium Systems: From Dissipative Structures to Order through fluctuations*. John Wiley & Sons, Inc. ISBN: 978-0-471-02401-9.
- NILSSON, U. und MAŁUSZYŃSKI, J. (1995). *Logic, programming, and Prolog*. 2. Aufl. John Wiley. ISBN: 978-0-471-95996-0.
- NOBELVERSAMMLUNG (2001). *Pressemitteilung: Den Nobelpreis des Jahres 2001 in Physiologie oder Medizin, 8.10.2001*. Nobelpreis des Jahres 2001 in Physiologie oder Medizin für Leland H. Hartwell, R. Timothy (Tim) Hunt und Paul M. Nurse gemeinsam für ihre Entdeckungen betreffend „Die Kontrolle des Zellzyklus“. URL: <http://nobelprize.org/medicine/laureates/2001/press-ge.html> (besucht am 30. 11. 2012).
- NOREL, R. und AGUR, Z. (1991). *A model for the adjustment of the mitotic clock by cyclin and MPF levels*. In: *Science* 251:4997 (März 1991), S. 1076–1078. DOI: [10.1126/science.1825521](https://doi.org/10.1126/science.1825521).
- NOWAK, M. A. und BANGHAM, C. R. M. (1996). *Population Dynamics of Immune Responses to Persistent Viruses*. In: *Science* 272:5258, S. 74–79. DOI: [10.1126/science.272.5258.74](https://doi.org/10.1126/science.272.5258.74).
- NOY, N. F. und MCGUINNESS, D. L. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*. Techn. Ber. KSL-01-05. Knowledge Systems Laboratory, Stanford University, USA, März 2001. URL: [http://www.ksl.stanford.edu/KSL\\_Abstracts/KSL-01-05.html](http://www.ksl.stanford.edu/KSL_Abstracts/KSL-01-05.html).
- ONO, N. und IKEGAMI, T. (1999). *Model of Self-Replicating Cell Capable of Self-Maintenance*. In: *Proceedings of the Fifth European Conference on Artificial Life (ECAL'99)*. Hrsg. von D. FLOREANO; J.-D. NICOUD und F. MONDANA. Bd. 1674. Lecture Notes in Computer Science. Berlin: Springer, S. 399–406.
- ORTH, J. D.; THIELE, I. und PALSSON, B. O. (2010). *What is flux balance analysis?* In: *Nature Biotechnology* 28:3 (März 2010), S. 245–248. DOI: [10.1038/nbt.1614](https://doi.org/10.1038/nbt.1614).
- OWSNICKI-KLEWE, B.; VON LUCK, K. und NEBEL, B. (2000). *Wissensrepräsentation und Logik - Eine Einführung*. In: *Handbuch der Künstlichen Intelligenz*. Hrsg. von G. GÖRZ; C.-R. ROLLINGER und J. SCHNEEBERGER. 3. Aufl. München, Wien: Oldenbourg Verlag. Kap. 5, S. 153–197.



- PASKIN, N. (2010). *Digital Object Identifier (DOI®) System*. In: *Encyclopedia of Library and Information Sciences, Third Edition*. 3. Aufl. Kap. 159, S. 1586–1592. DOI: [10.1081/E-ELIS3-120044418](https://doi.org/10.1081/E-ELIS3-120044418).
- PATASHNIK, O. (1988). *BibTeXing*. Comprehensive TeX Archive Network (CTAN). 8. Feb. 1988. URL: <http://www.ctan.org/tex-archive/biblio/bibtex/base/btxdoc.pdf>.
- PESCHL, M. F. und RIEGLER, A. (1999). *Does Representation Need Reality? Rethinking Epistemological Issues in the Light of Recent Developments and Concepts in Cognitive Science*. In: *Understanding Representation in the Cognitive Sciences: Does Representation Need Reality?* Hrsg. von A. RIEGLER; M. F. PESCHL und A. VON STEIN. New York: Kluwer Academic/Plenum, S. 9–17. DOI: [10.1007/978-0-585-29605-0\\_1](https://doi.org/10.1007/978-0-585-29605-0_1).
- PUPPE, F.; STOYAN, H. und STUDER, R. (2000). *Knowledge Engineering*. In: *Handbuch der Künstlichen Intelligenz*. Hrsg. von G. GÖRZ; C.-R. ROLLINGER und J. SCHNEEBERGER. 3. Aufl. München, Wien: Oldenbourg Verlag. Kap. 15, S. 599–641.
- QUINE, W. V. (1948). *On What There Is*. In: *Review of Metaphysics* 2 (Jan. 1948), S. 21–38.
- RESCHER, N. (1954). *Review of Jonathan Cohen: On the project of a universal character (Mind 1954, Volume 63, Issue 249, 49–63)*. In: *The Journal of Symbolic Logic* 19:2, S. 133. URL: <http://www.jstor.org/stable/2268890>.
- ROBINSON, P. und MUNDLOS, S. (2010). *The Human Phenotype Ontology*. In: *Clinical Genetics* 77:6, S. 525–534. DOI: [10.1111/j.1399-0004.2010.01436.x](https://doi.org/10.1111/j.1399-0004.2010.01436.x).
- RODRIGUEZ, N.; DONIZELLI, M. und LE NOVÈRE, N. (2007). *SBMLeditor: effective creation of models in the Systems Biology Markup Language (SBML)*. In: *BMC Bioinformatics* 8:1, S. 79. DOI: [10.1186/1471-2105-8-79](https://doi.org/10.1186/1471-2105-8-79).
- ROSEN, R. (1985). *Anticipatory systems*. Oxford, UK: Pergamon Press. ISBN: 978-0-08-031158-6.
- ROSEN, R. (1991). *Life itself*. New York: Columbia University Press. ISBN: 978-0-231-07564-0.
- RUST, H. (2005). *Operational Semantics for Timed Systems: A Non-standard Approach to Uniform Modeling of Timed and Hybrid Systems*. Lecture Notes in Computer Science 3456. Springer. ISBN: 978-3-540-25576-5.
- SAYERS, E. W.; BARRETT, T.; BENSON, D. A.; BRYANT, S. H.; CANESE, K.; CHETVERNIN, V.; CHURCH, D. M.; DICUCCIO, M.; EDGAR, R.; FEDERHEN, S.; FEOLO, M.; GEER, L. Y.; HELMBERG, W.; KAPUSTIN, Y.; LANDSMAN, D.; LIPMAN, D. J.; MADDEN, T. L.; MAGLOTT, D. R.; MILLER, V.; MIZRACHI, I.; OSTELL, J.; PRUITT, K. D.; SCHULER, G. D.; SEQUEIRA, E.; SHERRY, S. T.; SHUMWAY, M.; SIROTKIN, K.; SOUVOROV, A.; STARCHENKO, G.; TATUSOVA, T. A.; WAGNER, L.; YASCHENKO, E. und YE, J. (2009). *Database resources of the National Center for Biotechnology Information*. In: *Nucleic Acids Research* 37:D1 (Jan. 2009): *Database issue*, S. D5–15. DOI: [10.1093/nar/gkn741](https://doi.org/10.1093/nar/gkn741).
- sbml-discuss*. Mailingliste. URL: [http://sbml.org/Forums/index.php?t=threadt&frm\\_id=2&rid=0](http://sbml.org/Forums/index.php?t=threadt&frm_id=2&rid=0).
- SCHAFF, J.; FINK, C. C.; SLEPCHENKO, B.; CARSON, J. H. und LOEW, L. M. (1997). *A general computational framework for modeling cellular structure and function*. In: *Biophysical Journal* 73:3 (Sep. 1997), S. 1135–1146. DOI: [10.1016/S0006-3495\(97\)78146-3](https://doi.org/10.1016/S0006-3495(97)78146-3).
- SCHAICH, D.; BECKER, R. und KING, R. (2001). *Qualitative modelling for automatic identification of mathematical models of chemical reaction systems*. In: *Control Engineer-*

- ring Practice 9:12 (Dez. 2001), S. 1373–1381. URL: <http://www.sciencedirect.com/science/article/B6V2H-44KDC12-C/2/64ed231cf4f3fbf3537dcb60ebef21fc>.
- SCHÄUFLER, C. (2012). *Konstruktive Ontologie – Ontologien zwischen Wirklichkeitsbeschreibung und -konstruktion*. In: *Logik und Geschichte – Beiträge aus Jena*. Hrsg. von W. KIENZLER und S. SCHLOTTER. Jena: TABVLA-RASA-Verlag Jena, S. 123–133. URL: <http://www.tabvlarasa.de/44/Schaeufler.pdf>.
- SCHOBER, D.; SMITH, B.; LEWIS, S.; KUSNIERCZYK, W.; LOMAX, J.; MUNGALL, C.; TAYLOR, C.; ROCCA-SERRA, P. und SANSONE, S.-A. (2009). *Survey-based naming conventions for use in OBO Foundry ontology development*. In: *BMC Bioinformatics* 10:1, S. 125. DOI: [10.1186/1471-2105-10-125](https://doi.org/10.1186/1471-2105-10-125).
- SCHÖNING, U. (2000). *Logik für Informatiker*. 5. Aufl. Spektrum Akademischer Verlag. ISBN: 3827410053.
- SED-ML-discuss. Mailingliste. URL: <https://lists.sourceforge.net/lists/listinfo/sed-ml-discuss/>.
- SHAFRANOVICH, Y. (2005). *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. RFC 4180 (Informational). Internet Engineering Task Force, Okt. 2005. URL: <http://www.ietf.org/rfc/rfc4180.txt>.
- SHIMOYAMA, M.; NIGAM, R.; MCINTOSH, L. S.; NAGARAJAN, R.; RICE, T.; RAO, D. C. und DWINELL, M. R. (2012). *Three ontologies to define phenotype measurement data*. In: *Front Genet* 3, S. 87–87. DOI: [10.3389/fgene.2012.00087](https://doi.org/10.3389/fgene.2012.00087).
- SIMON, H. A. (1969). *The Sciences of the Artificial*. Cambridge, MA: MIT Press.
- SMITH, B.; WILLIAMS, J. und SCHULZE-KREMER, S. (2003). *The ontology of the gene ontology*. In: *Proc AMIA Symp*, S. 609–613.
- SMITH, B. (2004). *Ontology*. In: *The Blackwell Guide to the Philosophy of Computing and Information (Blackwell Philosophy Guides)*. Hrsg. von L. FLORIDI. Malden, Oxford, Carlton: Blackwell Publishing. Kap. 11, S. 155–166.
- SMITH, B. (2008). *Realitätsrepräsentation: Das Ziel der Ontologie*. In: *Biomedizinische Ontologie: Wissen strukturieren für den Informatik-Einsatz*. Hrsg. von L. JANSEN und B. SMITH. vdf Hochschulverlag AG an der ETH Zürich. Kap. 2. URL: <http://books.google.de/books?id=15NRG0ygx78C>.
- SMITH, B.; ASHBURNER, M.; ROSSE, C.; BARD, J.; BUG, W.; CEUSTERS, W.; GOLDBERG, L. J.; EILBECK, K.; IRELAND, A.; MUNGALL, C. J.; THE OBI CONSORTIUM; LEONTIS, N.; ROCCA-SERRA, P.; RUTTENBERG, A.; SANSONE, S.-A.; SCHEUERMANN, R. H.; SHAH, N.; WHETZEL, P. L. und LEWIS, S. (2007). *The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration*. In: *Nature Biotechnology* 25:11, S. 1251–1255. DOI: [10.1038/nbt1346](https://doi.org/10.1038/nbt1346).
- SMITH, B.; CEUSTERS, W.; KLAGGES, B.; KOHLER, J.; KUMAR, A.; LOMAX, J.; MUNGALL, C.; NEUHAUS, F.; RECTOR, A. und ROSSE, C. (2005). *Relations in biomedical ontologies*. In: *Genome Biology* 6:5, R46. DOI: [10.1186/gb-2005-6-5-r46](https://doi.org/10.1186/gb-2005-6-5-r46).
- SMITH, B. und KLAGGES, B. (2008). *Philosophie und biomedizinische Forschung*. In: *Biomedizinische Ontologie: Wissen strukturieren für den Informatik-Einsatz*. Hrsg. von L. JANSEN und B. SMITH. vdf Hochschulverlag AG an der ETH Zürich. Kap. 1. URL: <http://books.google.de/books?id=15NRG0ygx78C>.
- SMITH, B.; KÖHLER, J. und KUMAR, A. (2004). *On the Application of Formal Principles to Life Science Data: a Case Study in the Gene Ontology*. In: *Data Integration in the Life Sciences, First International Workshop (DILS 2004), Proceedings*. Hrsg. von

- E. RAHM. Bd. 1994. *Lecture Notes in Computer Science*. Springer, S. 79–94. DOI: [10.1007/978-3-540-24745-6\\_6](https://doi.org/10.1007/978-3-540-24745-6_6).
- SMITH, B. und WELTY, C. (2001). *FOIS introduction: Ontology – towards a new synthesis*. In: *Proceedings of the international conference on Formal Ontology in Information Systems*. Ogunquit, Maine, USA: ACM Press, S. iii–x. DOI: <http://doi.acm.org/10.1145/505168.505201>.
- SMITH, B. C. (1982). *Procedural Reflection in Programming Languages*. Diss. Massachusetts Institute of Technology. URL: <http://publications.csail.mit.edu/lcs/specpub.php?id=840>.
- SMITH, L. P.; BERGMANN, F. T.; CHANDRAN, D. und SAURO, H. M. (2009). *Antimony: a modular model definition language*. In: *Bioinformatics* 25:18, S. 2452–2454. DOI: [10.1093/bioinformatics/btp401](https://doi.org/10.1093/bioinformatics/btp401).
- SMOLEN, P.; BAXTER, D. A. und BYRNE, J. H. (2000). *Modeling Transcriptional Control in Gene Networks – Methods, Recent Results, and Future Directions*. In: *Bulletin of Mathematical Biology* 62:2, S. 247–292.
- SNOEP, J. L. und OLIVIER, B. G. (2002). *Java Web Simulation (JWS); a web based database of kinetic models*. In: *Molecular biology reports* 29:1-2, S. 259–263. URL: <http://www.ncbi.nlm.nih.gov/pubmed/12241068>.
- SNOEP, J. L. (2005). *The Silicon Cell initiative: working towards a detailed kinetic description at the cellular level*. In: *Current Opinion in Biotechnology* 16:3: *Environmental biotechnology/Systems biology*, S. 336–343. DOI: [10.1016/j.copbio.2005.05.003](https://doi.org/10.1016/j.copbio.2005.05.003).
- SOWA, J. F. (2000a). *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Pacific Grove, CA: Brooks Cole Publishing Co. ISBN: 978-0-534-94965-5.
- SOWA, J. F. (2000b). *Ontology, Metadata, and Semiotics*. In: *Proceedings of the Linguistic on Conceptual Structures*. Guided Tour of Ontology, Part II. Springer-Verlag, S. 55–81.
- SOWA, J. F. (2001). *KR Ontology*. Guided Tour of Ontology, Part IV. URL: <http://www.jfsowa.com/ontology/kronto.htm>.
- SPRIET, J. A. und VANSTEENKISTE, G. C. (1982). *Computer-aided modelling and simulation*. London; New York: Academic Press. ISBN: 978-0-12-659050-0.
- STACHOWIAK, H. (1973). *Allgemeine Modelltheorie*. Wien, New York: Springer-Verlag. ISBN: 978-3-211-81106-1.
- STEVENS, R.; HORROCKS, I.; GOBLE, C. und BECHHOFFER, S. (2001). *Building a Reasonable Bioinformatics Ontology Using OIL*. In: *Proceedings of the IJCAI-2001 Workshop on Ontologies and Information Sharing*. CEUR Workshop Proceedings 47. Sun SITE Central Europe, RWTH Aachen, S. 81–90. URL: <http://ceur-ws.org/Vol-47/stevens.pdf>.
- STOCK, W. (2009). *Begriffe und semantische Relationen in der Wissensrepräsentation*. In: *Information - Wissenschaft und Praxis* 60:8, S. 403–420.
- STROGATZ, S. H. (2001). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Perseus Books Group. ISBN: 978-0-7382-0453-6.
- STRUSS, P. (2000). *Modellbasierte Systeme und qualitative Modellierung*. In: *Handbuch der Künstlichen Intelligenz*. Hrsg. von G. GÖRZ; C.-R. ROLLINGER und J. SCHNEEBERGER. 3. Aufl. München, Wien: Oldenbourg Verlag.
- SUPPES, P. E. (1960). *A comparison of the meaning and uses of models in mathematics and the empirical sciences*. In: *Synthese* 12:2 (Sep. 1960), S. 287–301. DOI: [10.1007/BF00485107](https://doi.org/10.1007/BF00485107).

- TAYLOR, C. F.; FIELD, D.; SANSONE, S.-A.; AERTS, J.; APWEILER, R.; ASHBURNER, M.; BALL, C. A.; BINZ, P.-A.; BOGUE, M.; BOOTH, T.; BRAZMA, A.; BRINKMAN, R. R.; MICHAEL CLARK, A.; DEUTSCH, E. W.; FIEHN, O.; FOSTEL, J.; GHAZAL, P.; GIBSON, F.; GRAY, T.; GRIMES, G.; HANCOCK, J. M.; HARDY, N. W.; HERMJAKOB, H.; JULIAN, R. K.; KANE, M.; KETTNER, C.; KINSINGER, C.; KOLKER, E.; KUIPER, M.; LE NOVÈRE, N.; LEEBENS-MACK, J.; LEWIS, S. E.; LORD, P.; MALLON, A.-M.; MARTHANDAN, N.; MASUYA, H.; McNALLY, R.; MEHRLE, A.; MORRISON, N.; ORCHARD, S.; QUACKENBUSH, J.; REECY, J. M.; ROBERTSON, D. G.; ROCCA-SERRA, P.; RODRIGUEZ, H.; ROSENFELDER, H.; SANTOYO-LOPEZ, J.; SCHEUERMANN, R. H.; SCHÖBER, D.; SMITH, B.; SNAPE, J.; STOECKERT, C. J.; TIPTON, K.; STERK, P.; UNTERGASSER, A.; VANDESOMPELE, J. und WIEMANN, S. (2008). *Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project*. In: *Nature Biotechnology* 26:8 (Aug. 2008), S. 889–896. DOI: [10.1038/nbt.1411](https://doi.org/10.1038/nbt.1411).
- THIELE, I.; SWAINSTON, N.; FLEMING, R. M. T.; HOPPE, A.; SAHOO, S.; AURICH, M. K.; HARALDSDOTTIR, H.; MO, M. L.; ROLFSSON, O.; STOBBE, M. D.; THORLEIFSSON, S. G.; AGREN, R.; BÖLLING, C.; BORDEL, S.; CHAVALI, A. K.; DOBSON, P.; DUNN, W. B.; ENDLER, L.; HALA, D.; HUCKA, M.; HULL, D.; JAMESON, D.; JAMSHIDI, N.; JONSSON, J. J.; JUTY, N.; KEATING, S.; NOOKAEW, I.; LE NOVÈRE, N.; MALYS, N.; MAZEIN, A.; PAPIN, J. A.; PRICE, N. D.; SR, E. S.; SIGURDSSON, M. I.; SIMEONIDIS, E.; SONNENSCHN, N.; SMALLBONE, K.; SOROKIN, A.; VAN BEEK, J. H. G. M.; WEICHART, D.; GORYANIN, I.; NIELSEN, J.; WESTERHOFF, H. V.; KELL, D. B.; MENDES, P. und PALSSON, B. Ø. (2013). *A community-driven global reconstruction of human metabolism*. In: *Nature Biotechnology* 31:5, S. 419–425. DOI: [10.1038/nbt.2488](https://doi.org/10.1038/nbt.2488).
- TRELEASE, R. B. und PARK, J. (1996). *Qualitative process modeling of cell-cell-pathogen interactions in the immune system*. In: *Computer Methods and Programs in Biomedicine* 51:3, S. 171–181. DOI: [10.1016/S0169-2607\(96\)01768-3](https://doi.org/10.1016/S0169-2607(96)01768-3).
- TUTTLE, M. und NELSON, S. (1996). *A poor precedent*. In: *Methods of Information in Medicine* 35:3 (Sep. 1996), S. 211–217.
- TYSON, J. und KAUFFMAN, S. (1975). *Control of mitosis by a continuous biochemical oscillator: synchronization and spatially inhomogeneous oscillations*. In: *Journal of Mathematical Biology* 1:4, S. 289–310. DOI: [10.1007/BF00279848](https://doi.org/10.1007/BF00279848).
- TYSON, J. (1991). *Modeling the cell division cycle: cdc2 and cyclin interactions*. In: *Proceedings of the National Academy of Sciences of the United States of America* 88:16 (Aug. 1991), S. 7328–7332. DOI: [10.1073/pnas.88.16.7328](https://doi.org/10.1073/pnas.88.16.7328).
- ULLAH, M. und WOLKENHAUER, O. (2011). *Stochastic approaches for systems biology*. New York: Springer. ISBN: 978-1-4614-0478-1.
- UMEDA, Y. und TOMIYAMA, T. (1995). *FBS modeling: modeling scheme of function for conceptual design*. In: *Working Papers of the Ninth International Workshop on Qualitative Reasoning about Physical Systems*, S. 271–278.
- USCHOLD, M. und JASPER, R. (1999). *A Framework for Understanding and Classifying Ontology Applications*. In: *Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods*. CEUR Workshop Proceedings 18. Sun SITE Central Europe, RWTH Aachen. URL: <http://ceur-ws.org/Vol-18/11-uschold.pdf>.
- USCHOLD, M. (1998). *Knowledge level modelling: concepts and terminology*. In: *The Knowledge Engineering Review* 13:01 (März 1998), S. 5–29. DOI: [10.1017/S0269888998001040](https://doi.org/10.1017/S0269888998001040).

- USCHOLD, M. und GRÜNINGER, M. (1996). *Ontologies: principles, methods, and applications*. In: *Knowledge Engineering Review* 11:2, S. 93–155.
- VAN HARMELEN, F.; LIFSCHITZ, V. und PORTER, B., Hrsg. (2008). *Handbook of Knowledge Representation*. 1. Aufl. Foundations of Artificial Intelligence 3. Elsevier Science. ISBN: 978-0-444-52211-5.
- VARZI, A. und VIEU, L., Hrsg. (2004). *Formal Ontology in Information Systems Proceedings of the Third International Conference (FOIS-2004)*. Frontiers in Artificial Intelligence and Applications 114. IOS Press, Okt. 2004. ISBN: 978-1-58603-468-9.
- WALTEMATH, D.; ADAMS, R.; BEARD, D. A.; BERGMANN, F. T.; BHALLA, U. S.; BRITTEN, R.; CHELLIAH, V.; COOLING, M. T.; COOPER, J.; CRAMPIN, E. J.; GARNY, A.; HOOPS, S.; HUCKA, M.; HUNTER, P.; KLIPP, E.; LAIBE, C.; MILLER, A. K.; MORARU, I.; NICKERSON, D.; NIELSEN, P.; NIKOLSKI, M.; SAHLE, S.; SAURO, H. M.; SCHMIDT, H.; SNOEP, J. L.; TOLLE, D.; WOLKENHAUER, O. und LE NOVÈRE, N. (2011a). *Minimum Information About a Simulation Experiment (MIASE)*. In: *PLOS Computational Biology* 7:4 (Apr. 2011), e1001122. DOI: [10.1371/journal.pcbi.1001122](https://doi.org/10.1371/journal.pcbi.1001122).
- WALTEMATH, D.; ADAMS, R.; BERGMANN, F.; HUCKA, M.; KOLPAKOV, F.; MILLER, A.; MORARU, I.; NICKERSON, D.; SAHLE, S.; SNOEP, J. und LE NOVÈRE, N. (2011b). *Reproducible computational biology experiments with SED-ML - The Simulation Experiment Description Markup Language*. In: *BMC Systems Biology* 5:1, S. 198. DOI: [10.1186/1752-0509-5-198](https://doi.org/10.1186/1752-0509-5-198).
- WALTEMATH, D.; HENKEL, R.; HÄLKE, R.; SCHARM, M. und WOLKENHAUER, O. (2013). *Improving the reuse of computational models through version control*. In: *Bioinformatics* 29:6, S. 742–748. DOI: [10.1093/bioinformatics/btt018](https://doi.org/10.1093/bioinformatics/btt018).
- WALTEMATH, D.; SWAINSTON, N.; LISTER, A.; BERGMANN, F.; HENKEL, R.; HOOPS, S.; HUCKA, M.; JUTY, N.; KEATING, S.; KNÜPFER, C.; KRAUSE, F.; LAIBE, C.; LIEBERMEISTER, W.; LLOYD, C.; MISIRLI, G.; SCHULZ, M.; TASCHUK, M. und LE NOVÈRE, N. (2011c). *SBML Level 3 Package Proposal: Annotation*. Available from Nature Precedings. DOI: [10.1038/npre.2011.5610.1](https://doi.org/10.1038/npre.2011.5610.1).
- WEBB, B. (2001). *Can robots make good models of biological behaviour?* In: *Behavioral and Brain Sciences* 24:06 (Dez. 2001), S. 1033–1050. DOI: [10.1017/S0140525X01000127](https://doi.org/10.1017/S0140525X01000127).
- WEDEKIND, H.; GÖRZ, G.; KÖTTER, R. und INHETVEEN, R. (1998). *Modellierung, Simulation, Visualisierung: Zu aktuellen Aufgaben der Informatik*. Zur Diskussion gestellt. In: *Informatik Spektrum* 21:5, S. 265–272. DOI: [10.1007/s002870050104](https://doi.org/10.1007/s002870050104).
- WHETZEL, P. L.; NOY, N. F.; SHAH, N. H.; ALEXANDER, P. R.; NYULAS, C.; TUDORACHE, T. und MUSEN, M. A. (2011). *BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications*. In: *Nucleic Acids Research* 39:suppl 2, W541–W545. DOI: [10.1093/nar/gkr469](https://doi.org/10.1093/nar/gkr469).
- WITTIG, U.; KANIA, R.; GOLEBIEWSKI, M.; REY, M.; SHI, L.; JONG, L.; ALGAA, E.; WEIDEMANN, A.; SAUER-DANZWITZ, H.; MIR, S.; KREBS, O.; BITTKOWSKI, M.; WETSCH, E.; ROJAS, I. und 0001, W. M. (2012). *SABIO-RK – database for biochemical reaction kinetics*. In: *Nucleic Acids Research* 40:D1 (Jan. 2012): Database issue, S. 790–796. DOI: [10.1093/nar/gkr1046](https://doi.org/10.1093/nar/gkr1046).
- WOLKENHAUER, O.; ULLAH, M.; WELLSTEAD, P. und CHO, K.-H. (2005). *The dynamic systems approach to control and regulation of intracellular networks*. In: *FEBS Letters* 579:8 (März 2005), S. 1846–1853. DOI: [10.1016/j.febslet.2005.02.008](https://doi.org/10.1016/j.febslet.2005.02.008).



- ZEIGLER, B. P. (1984). *Multifaceted modelling and discrete event simulation*. London; Orlando: Academic Press. ISBN: 978-0-12-778450-2.
- ZEIGLER, B. P.; PRAEHOFFER, H. und KIM, T. G. (2000). *Theory of Modeling and Simulation*. 2. Aufl. Academic Press. ISBN: 978-0-12-778455-7.
- ZIEMKE, T. (1999). *Rethinking Grounding*. In: *Understanding Representation in the Cognitive Sciences*. Hrsg. von A. RIEGLER; M. PESCHL und A. VON STEIN. New York: Plenum Press, S. 177–190.
- ZIMMERLI, W. C. und WOLF, S., Hrsg. (1994). *Künstliche Intelligenz: Philosophische Probleme*. Stuttgart: Reclam. ISBN: 978-3-15-008922-4.

# Anhang





## A. Abkürzungen und Projekte

Im Folgenden sind verwendete Abkürzungen sowie referenzierte Projekte (MI-Standards, formale Sprachen, Ontologien, Werkzeuge usw.) aufgelistet.

<b>ABox</b>	<i>Assertional Box</i> (Fakten-Teil einer Wissensbasis)
<b>Antimony</b>	Modellierungssprache, SMITH ET AL. 2009, <a href="http://antimony.sourceforge.net/">http://antimony.sourceforge.net/</a>
<b>ATP</b>	Adenosintriphosphat
<b>BFO</b>	<i>Basic Formal Ontology</i> , GRENON ET AL. 2004, <a href="http://www.ifomis.org/bfo/">http://www.ifomis.org/bfo/</a>
<b>BibTeX</b>	Literaturverwaltungsprogramm, PATASHNIK 1988
<b>BIOCHAM</b>	<i>Biochemical Abstract Machine</i> , CHABRIER-RIVIER ET AL. 2005
<b>BioModels Database</b>	Modelldatenbank, LE NOVÈRE ET AL. 2006, <a href="http://www.ebi.ac.uk/biomodels/">http://www.ebi.ac.uk/biomodels/</a>
<b>BioModels.net</b>	Community, LE NOVÈRE 2006a, <a href="http://biomodels.net/">http://biomodels.net/</a>
<b>BioModels.net Qualifiers</b>	Sammlung von Qualifikatoren für Bio-Modelle, <a href="http://co.mbine.org/standards/qualifiers">http://co.mbine.org/standards/qualifiers</a>
<b>BioPAX</b>	<i>Biological Pathways Exchange</i> , DEMIR ET AL. 2010, <a href="http://www.biopax.org/">http://www.biopax.org/</a>
<b>BioPortal</b>	Internetportal für Ontologien, WHETZEL ET AL. 2011, <a href="http://bioportal.bioontology.org/">http://bioportal.bioontology.org/</a>
<b>cdc2</b>	Synonym für CDK1
<b>CDK1</b>	<i>Cyclin-dependent Kinase 1</i>
<b>CellDesigner</b>	Simulator, FUNAHASHI ET AL. 2003, <a href="http://celldesigner.org/">http://celldesigner.org/</a>
<b>CellML</b>	Modellierungssprache, CUELLAR ET AL. 2003, <a href="http://www.cellml.org/">http://www.cellml.org/</a>
<b>CellML Model Repository</b>	Modelldatenbank, LLOYD ET AL. 2008, <a href="http://www.cellml.org/models">http://www.cellml.org/models</a>
<b>ChEBI</b>	<i>Chemical Entities of Biological Interest</i> , DEGTYARENKO ET AL. 2008, <a href="http://www.ebi.ac.uk/chebi">http://www.ebi.ac.uk/chebi</a>
<b>CL</b>	<i>Cell Ontology</i> , BARD ET AL. 2005, <a href="http://cellontology.org">http://cellontology.org</a>
<b>CLP</b>	<i>Constraint Logic Programming</i> , siehe z.B. APT 2003
<b>COMBINE</b>	<i>COmputational Modeling in BIOlogy NEtwork</i> , Community, <a href="http://co.mbine.org/">http://co.mbine.org/</a>
<b>COPASI</b>	<i>COmplex PATHway SIimulator</i> , Simulator, HOOFS ET AL. 2006, <a href="http://www.copasi.org/">http://www.copasi.org/</a>
<b>CSV</b>	<i>Comma-Separated Values</i> , SHAFRANOVICH 2005, <a href="http://www.ietf.org/rfc/rfc4180.txt">http://www.ietf.org/rfc/rfc4180.txt</a>
<b>CTL</b>	<i>Computation Tree Logic</i> , eingeschränkte Version, temporale Logik
<b>CTL*</b>	<i>Computation Tree Logic</i> , temporale Logik

<b>CVODE</b>	numerischer Integrationsalgorithmus, <a href="#">HINDMARSH ET AL. 2005</a> , <a href="http://computation.llnl.gov/casc/sundials/">http://computation.llnl.gov/casc/sundials/</a>
<b>CYC</b>	Wissensbasis, <a href="#">LENAT ET AL. 1990</a>
<b>DOI</b>	<i>Digital Object Identifier</i> , <a href="#">PASKIN 2010</a> , <a href="http://www.doi.org/">http://www.doi.org/</a>
<b>Dublin Core</b>	Standard für Metainformationen, <a href="#">KUNZE ET AL. 2007</a> , <a href="http://dublincore.org/">http://dublincore.org/</a>
<b>DYML</b>	<i>Dynamics Markup Language</i> , <a href="#">KNÜPFER ET AL. 2013b</a>
<b>EBI</b>	<i>European Bioinformatics Institute</i> , Hinxton/Cambridge, Großbritannien, <a href="http://www.ebi.ac.uk/">http://www.ebi.ac.uk/</a>
<b>ECO</b>	<i>Evidence Codes Ontology</i> , <a href="http://www.evidenceontology.org">http://www.evidenceontology.org</a>
<b>EcoCyc</b>	Datenbank für <i>Escherichia coli</i> , <a href="#">KESELER ET AL. 2013</a> , <a href="http://www.ecocyc.org/">http://www.ecocyc.org/</a>
<b>EngMath</b>	<i>Engineering Mathematics Ontology</i> , <a href="#">GRUBER ET AL. 1994</a>
<b>ENVISION</b>	qualitative Modellierungsmethode, <a href="#">KLEER ET AL. 1984</a>
<b>FaCT++</b>	OWL-Inferenzmaschine, <a href="http://owl.man.ac.uk/factplusplus/">http://owl.man.ac.uk/factplusplus/</a>
<b>Fielded Text</b>	Dateiformat für numerische Daten, <a href="http://www.fieldedtext.org/">http://www.fieldedtext.org/</a>
<b>FOL</b>	Prädikatenlogik erster Stufe (von <i>First-Order Logic</i> )
<b>Fortran</b>	Programmiersprache, <a href="http://www.nag.co.uk/sc22wg5/">http://www.nag.co.uk/sc22wg5/</a>
<b>FuGE</b>	<i>Functional Genomics Experiment Object Model</i> , <a href="#">JONES ET AL. 2007</a> , <a href="http://fuge.sourceforge.net/">http://fuge.sourceforge.net/</a>
<b>Garp3</b>	Werkzeug für qualitative Modellierung und Simulation, <a href="#">BREDEWEG ET AL. 2009</a> , <a href="http://hcs.science.uva.nl/QRM/">http://hcs.science.uva.nl/QRM/</a>
<b>GenBank</b>	Datenbank für DNA-Sequenzen, <a href="#">BENSON ET AL. 2013</a> , <a href="http://www.ncbi.nlm.nih.gov/genbank/">http://www.ncbi.nlm.nih.gov/genbank/</a>
<b>GNU Octave</b>	Programmiersprache, <a href="http://gnu.org/software/octave">http://gnu.org/software/octave</a>
<b>GO</b>	<i>Gene Ontology</i> , <a href="#">GENE ONTOLOGY CONSORTIUM 2001</a>
<b>HermiT</b>	OWL-Inferenzmaschine, <a href="#">MOTIK ET AL. 2009</a> , <a href="http://hermit-reasoner.com/">http://hermit-reasoner.com/</a>
<b>HPO</b>	<i>Human Phenotype Ontology</i> , <a href="#">ROBINSON ET AL. 2010</a> , <a href="http://www.human-phenotype-ontology.org/">http://www.human-phenotype-ontology.org/</a>
<b>ICD</b>	<i>International Classification of Diseases</i> , <a href="http://www.who.int/classifications/icd/">http://www.who.int/classifications/icd/</a>
<b>identifiers.org</b>	Resolver für URIs, <a href="#">JUTY ET AL. 2012</a> , <a href="http://identifiers.org/">http://identifiers.org/</a>
<b>IntEnz</b>	<i>Integrated relational Enzyme database</i> , <a href="#">FLEISCHMANN ET AL. 2004</a> , <a href="http://www.ebi.ac.uk/intenz/">http://www.ebi.ac.uk/intenz/</a>
<b>IRI</b>	<i>Internationalized Resource Identifier</i>
<b>JWS Online</b>	<i>Java Web Simulation Online</i> , <a href="#">SNOEP ET AL. 2002</a> , <a href="http://jjj.biochem.sun.ac.za/">http://jjj.biochem.sun.ac.za/</a>
<b>KIF</b>	<i>Knowledge Interchange Format</i> , <a href="#">GENESERITH ET AL. 1992</a> , <a href="http://www-ksl.stanford.edu/knowledge-sharing/kif/">http://www-ksl.stanford.edu/knowledge-sharing/kif/</a>

<b>KiSAO</b>	<i>Kinetic Simulation Algorithm Ontology</i> , COURTOT ET AL. 2011, <a href="http://biomodels.net/kisao/">http://biomodels.net/kisao/</a>
<b>libSBML</b>	Programmbibliothek für SBML, BORNSTEIN ET AL. 2008, <a href="http://sbml.org/Software/libSBML">http://sbml.org/Software/libSBML</a>
<b>Lisp</b>	Programmiersprache, ABELSON ET AL. 1996
<b>LSODA</b>	<i>Livermore Solver for Ordinary Differential Equations with Automatic Method Selection</i> , numerischer Integrationsalgorithmus, HINDMARSH 1983
<b>LTL</b>	<i>Linear Temporal Logic</i> , temporale Logik
<b>Manchester Syntax</b>	Serialisierung für OWL, HORRIDGE ET AL. 2012
<b>MathML</b>	<i>Mathematical Markup Language</i> , AUSBROOKS ET AL. 2003
<b>MATLAB</b>	MATrix LABoratory, Programmiersprache, <a href="http://www.mathworks.de/">http://www.mathworks.de/</a>
<b>MIAME</b>	<i>Minimum Information About a Microarray Experiment</i> , BRAZMA ET AL. 2001
<b>MIASE</b>	<i>Minimum Information About a Simulation Experiment</i> , WALTEMATH ET AL. 2011a, <a href="http://co.mbine.org/standards/miase">http://co.mbine.org/standards/miase</a>
<b>MIBBI</b>	<i>Minimum Information for Biological and Biomedical Investigations</i> , LE NOVÈRE ET AL. 2005, <a href="http://biosharing.org/standards/mibbi">http://biosharing.org/standards/mibbi</a>
<b>MIFlowCyt</b>	<i>Minimum Information about a Flow Cytometry Experiment</i> , LEE ET AL. 2008
<b>MINTENTO</b>	<i>Modelling Intention Ontology</i> , KNÜPFER ET AL. 2013a
<b>MIRIAM</b>	<i>Minimal Information Required In the Annotation of Models</i> , LE NOVÈRE ET AL. 2005, <a href="http://co.mbine.org/standards/miriam">http://co.mbine.org/standards/miriam</a>
<b>MIRIAM Identifiers</b>	Identifikatoren für Bio-Modellierung, JUTY ET AL. 2012
<b>MIRIAM Registry</b>	Datenbank für ID-Namensräume, JUTY ET AL. 2012
<b>MI-Standard</b>	minimaler Informations-Standard
<b>MMO</b>	<i>Measurement Method Ontology</i> , SHIMOYAMA ET AL. 2012, <a href="http://phenoonto.sourceforge.net/">http://phenoonto.sourceforge.net/</a>
<b>MPF</b>	<i>Maturation Promoting Factor</i>
<b>NC-IUBMB</b>	<i>International Union of Biochemistry and Molecular Biology</i> , <a href="http://www.iubmb.org/">http://www.iubmb.org/</a>
<b>NuML</b>	<i>Numerical Markup Language</i> , <a href="http://code.google.com/p/numl/">http://code.google.com/p/numl/</a>
<b>OBO Format</b>	Spezifikationsformat für Ontologien, <a href="http://www.geneontology.org/G0.format.obo-1_2.shtml">http://www.geneontology.org/G0.format.obo-1_2.shtml</a>
<b>OBO Foundry</b>	<i>The Open Biological and Biomedical Ontologies Foundry</i> , SMITH ET AL. 2007, <a href="http://obofoundry.org/">http://obofoundry.org/</a>
<b>OBO-Edit</b>	Ontologie-Editor für das OBO Format, <a href="http://oboedit.org/">http://oboedit.org/</a>
<b>OLS</b>	<i>Ontology Lookup Service</i> , <a href="http://www.ebi.ac.uk/ontology-lookup/">http://www.ebi.ac.uk/ontology-lookup/</a>
<b>OPB</b>	<i>Ontology of Physics for Biology</i> , COOK ET AL. 2008
<b>ORCA</b>	<i>Ontology of Reasoning, Certainty and Attribution</i> , DE WAARD ET AL. 2012, <a href="http://vocab.deri.ie/orca">http://vocab.deri.ie/orca</a>

<b>ORCID</b>	<i>Open Researcher and Contributor ID</i> , BUTLER 2012, <a href="http://orcid.org/">http://orcid.org/</a>
<b>OWL</b>	<i>Web Ontology Language</i> , HITZLER ET AL. 2012, <a href="http://www.w3.org/OWL/">http://www.w3.org/OWL/</a>
<b>PATO</b>	<i>Phenotypic Quality Ontology</i> , GKOUTOS ET AL. 2004, <a href="http://obofoundry.org/wiki/index.php/PATO:Main_Page">http://obofoundry.org/wiki/index.php/PATO:Main_Page</a>
<b>Pellet</b>	OWL-Inferenzmaschine, <a href="http://clarkparsia.com/pellet">http://clarkparsia.com/pellet</a>
<b>PhysSys</b>	<i>Ontology for Physical Systems</i> , BORST ET AL. 1997
<b>PlanetMath</b>	Online-Enzyklopädie, <a href="http://planetmath.org/">http://planetmath.org/</a>
<b>PMID</b>	<i>PubMed Identifier</i>
<b>PO</b>	<i>Plant Ontology</i> , COOPER ET AL. 2013, <a href="http://www.plantontology.org/">http://www.plantontology.org/</a>
<b>Prolog</b>	Programmiersprache, NILSSON ET AL. 1995
<b>Protégé</b>	Entwicklungsumgebung für Ontologien, <a href="http://protege.stanford.edu/">http://protege.stanford.edu/</a>
<b>PubMed</b>	Publikationsdatenbank, <a href="http://www.ncbi.nlm.nih.gov/pubmed/">http://www.ncbi.nlm.nih.gov/pubmed/</a>
<b>PURL</b>	<i>Persistent Uniform Resource Locator</i>
<b>QDE</b>	<i>Qualitative Differential Equation</i>
<b>QPT</b>	<i>Qualitative Process Theory</i> , qualitative Modellierungsmethode, FORBUS 1984
<b>QSIM</b>	qualitative Modellierungsmethode, KUIPERS 1986, <a href="http://www.cs.utexas.edu/users/qr/">http://www.cs.utexas.edu/users/qr/</a>
<b>R</b>	Programmiersprache, <a href="http://www.r-project.org/">http://www.r-project.org/</a>
<b>RDF</b>	<i>Resource Description Framework</i> , MANOLA ET AL. 2004, <a href="http://www.w3.org/RDF/">http://www.w3.org/RDF/</a>
<b>RDF/XML</b>	XML-basierte Serialisierung für OWL ( <i>Web Ontology Language</i> , HITZLER ET AL. 2012), BECKETT 2004, <a href="http://www.w3.org/TR/rdf-syntax-grammar/">http://www.w3.org/TR/rdf-syntax-grammar/</a>
<b>RDFS</b>	<i>Resource Description Framework Schema</i> , GUHA ET AL. 2004, <a href="http://www.w3.org/TR/rdf-schema/">http://www.w3.org/TR/rdf-schema/</a>
<b>Reactome</b>	Datenbank für biologische Stoffwechsel- und Signalwege, JOSHI-TOPE ET AL. 2005
<b>Recon 2</b>	Modell des menschlichen Stoffwechsels, THIELE ET AL. 2013, <a href="http://humanmetabolism.org">http://humanmetabolism.org</a>
<b>RIS</b>	<i>Research Information System Format</i> , <a href="http://www.refman.com/support/risformat_intro.asp">http://www.refman.com/support/risformat_intro.asp</a>
<b>RO</b>	<i>OBO Relations Ontology</i> , <a href="http://code.google.com/p/obo-relations/">http://code.google.com/p/obo-relations/</a>
<b>SABIO-RK</b>	<i>System for the Analysis of Biochemical Pathways – Reaction Kinetics</i> , WITTIG ET AL. 2012
<b>Saint</b>	Annotationswerkzeug für Bio-Modelle, LISTER ET AL. 2009, <a href="http://saint-annotate.sourceforge.net">http://saint-annotate.sourceforge.net</a>
<b>SBGN</b>	<i>Systems Biology Graphical Notation</i> , LE NOVÈRE ET AL. 2009, <a href="http://www.sbgn.org">http://www.sbgn.org</a>
<b>SBGN-ED</b>	SBGN-Editor, Add-on für VANTED, CZAUDERNA ET AL. 2010, <a href="http://vanted.ipk-gatersleben.de/addons/sbgn-ed/">http://vanted.ipk-gatersleben.de/addons/sbgn-ed/</a>

<b>SBGN-PD</b>	SBGN Process Description Language, MOODIE ET AL. 2009, <a href="http://identifiers.org/combine.specifications/sbgn.pd.level-1.version-1.3">http://identifiers.org/combine.specifications/sbgn.pd.level-1.version-1.3</a>
<b>SBML</b>	<i>Systems Biology Markup Language</i> , HUCKA ET AL. 2003, <a href="http://sbml.org/">http://sbml.org/</a>
<b>SBML Annotation Package</b>	Erweiterung für SBML, WALTEMATH ET AL. 2011c
<b>SBML Test Suite</b>	Testumgebung für SBML, <a href="http://sbml.org/Software/SBML_Test_Suite">http://sbml.org/Software/SBML_Test_Suite</a>
<b>SBMLEditor</b>	Editor für SBML, RODRIGUEZ ET AL. 2007, <a href="http://www.ebi.ac.uk/compneur-srv/SBMLEditor.html">http://www.ebi.ac.uk/compneur-srv/SBMLEditor.html</a>
<b>SBMLsqueezer</b>	Werkzeug für kinetische Gleichungen, DRAGER ET AL. 2008, <a href="http://www.ra.cs.uni-tuebingen.de/software/SBMLsqueezer">http://www.ra.cs.uni-tuebingen.de/software/SBMLsqueezer</a>
<b>SBO</b>	<i>Systems Biology Ontology</i> , COURTOT ET AL. 2011, <a href="http://biomodels.net/sbo/">http://biomodels.net/sbo/</a>
<b>SBRML</b>	<i>Systems Biology Results Markup Language</i> , DADA ET AL. 2010, <a href="http://www.comp-sys-bio.org/SBRML">http://www.comp-sys-bio.org/SBRML</a>
<b>SBW</b>	<i>Systems Biology Workbench</i> , HUCKA ET AL. 2002
<b>SED-ML</b>	<i>Simulation Experiment Description Markup Language</i> , WALTEMATH ET AL. 2011b, <a href="http://sed-ml.org">http://sed-ml.org</a>
<b>SED-ML Web Tools</b>	Online Simulator, <a href="http://sysbioapps.dyndns.org/SED-ML_Web_Tools/">http://sysbioapps.dyndns.org/SED-ML_Web_Tools/</a>
<b>semanticSBML</b>	Annotationswerkzeug für Bio-Modelle, KRAUSE ET AL. 2010, <a href="http://semanticsbml.org/">http://semanticsbml.org/</a>
<b>SemSim</b>	<i>Semantic Simulation</i> , NEAL ET AL. 2009
<b>Silicon Cell</b>	systembiologisches Modellierungsprojekt, SNOEP 2005, <a href="http://www.siliconcell.net">http://www.siliconcell.net</a>
<b>SKOS</b>	<i>Simple Knowledge Organization System</i> , MILES ET AL. 2009, <a href="http://www.w3.org/TR/skos-reference/">http://www.w3.org/TR/skos-reference/</a>
<b>SourceForge</b>	Internetportal für Softwareprojekte, <a href="http://sourceforge.net/">http://sourceforge.net/</a>
<b>SPARQL</b>	<i>SPARQL Protocol And RDF Query Language</i> , HARRIS ET AL. 2013, <a href="http://www.w3.org/sparql/">www.w3.org/sparql/</a>
<b>SUNDIALS</b>	Bibliothek numerischer Integrationsalgorithmen, HINDMARSH ET AL. 2005, <a href="http://computation.llnl.gov/casc/sundials/">http://computation.llnl.gov/casc/sundials/</a>
<b>SVN</b>	<i>Apache Subversion</i> , Versionsverwaltung, <a href="http://subversion.apache.org/">http://subversion.apache.org/</a>
<b>SWRL</b>	<i>Semantic Web Rule Language</i> , HORROCKS ET AL. 2004, <a href="http://www.w3.org/Submission/SWRL/">http://www.w3.org/Submission/SWRL/</a>
<b>Taxonomy Database</b>	Datenbank für biologische Taxonomie und Nomenklatur, SAYERS ET AL. 2009, <a href="http://www.ncbi.nlm.nih.gov/taxonomy/">http://www.ncbi.nlm.nih.gov/taxonomy/</a>
<b>TBox</b>	<i>Terminological Box</i> (terminologischer Teil einer Wissensbasis)
<b>TEDDY</b>	<i>TErminology for the Description of DYnamics</i> , COURTOT ET AL. 2011, <a href="http://co.mbine.org/specifications/teddy">http://co.mbine.org/specifications/teddy</a>
<b>Turtle</b>	<i>Terse RDF Triple Language</i> , BECKETT ET AL. 2013, <a href="http://www.w3.org/TR/turtle/">http://www.w3.org/TR/turtle/</a>
<b>UniProt</b>	Datenbank für Proteine, APWEILER ET AL. 2004
<b>UO</b>	<i>Units Ontology</i> , GKOUTOS ET AL. 2012
<b>URI</b>	<i>Uniform Resource Identifier</i>

<b>URL</b>	<i>Uniform Resource Locator</i>
<b>URN</b>	<i>Uniform Resource Name</i>
<b>VANTED</b>	Visualization and Analysis of Networks containing Experimental Data, JUNKER ET AL. 2006, <a href="http://vanted.ipk-gatersleben.de">http://vanted.ipk-gatersleben.de</a>
<b>W3C</b>	<i>World Wide Web Consortium</i> , <a href="http://www.w3.org/">http://www.w3.org/</a>
<b>WebProtégé</b>	Entwicklungsumgebung für Ontologien, <a href="http://protegewiki.stanford.edu/wiki/WebProtege">http://protegewiki.stanford.edu/wiki/WebProtege</a>
<b>Wolfram SystemModeler</b>	Umgebung für Modellierung und Simulation, <a href="http://wolfram.com/system-modeler/">http://wolfram.com/system-modeler/</a>
<b>XAO</b>	<i>Xenopus Anatomy and Development Ontology</i>
<b>XCO</b>	<i>Experimental Conditions Ontology</i> , SHIMOYAMA ET AL. 2012, <a href="http://phenoonto.sourceforge.net/">http://phenoonto.sourceforge.net/</a>
<b>XML</b>	<i>Extensible Markup Language</i> , BRAY ET AL. 2006, <a href="http://www.w3.org/XML">http://www.w3.org/XML</a>
<b>XPath</b>	<i>XML Path Language</i> , BERGLUND ET AL. 2010, <a href="http://www.w3.org/Style/XSL/">http://www.w3.org/Style/XSL/</a>



## B. ID-Verzeichnis

Dieser Anhang enthält alle im Text referenzierten Datenbankeinträge und Ontologietерme mit der jeweiligen ID und der zugehörigen [URI](#) (*Uniform Resource Identifier*).

1831270	<a href="http://identifiers.org/pubmed/1831270">http://identifiers.org/pubmed/1831270</a>
33208	<a href="http://identifiers.org/taxonomy/33208">http://identifiers.org/taxonomy/33208</a>
33511	<a href="http://identifiers.org/taxonomy/33511">http://identifiers.org/taxonomy/33511</a>
6072	<a href="http://identifiers.org/taxonomy/6072">http://identifiers.org/taxonomy/6072</a>
7625	<a href="http://identifiers.org/taxonomy/7625">http://identifiers.org/taxonomy/7625</a>
BF0_0000015	<a href="http://purl.obolibrary.org/obo/BF0_0000015">http://purl.obolibrary.org/obo/BF0_0000015</a>
BF0_0000034	<a href="http://purl.obolibrary.org/obo/BF0_0000034">http://purl.obolibrary.org/obo/BF0_0000034</a>
BF0_0000066	<a href="http://purl.obolibrary.org/obo/BF0_0000066">http://purl.obolibrary.org/obo/BF0_0000066</a>
BIOMD0000000005	<a href="http://identifiers.org/biomodels.db/BIOMD0000000005">http://identifiers.org/biomodels.db/BIOMD0000000005</a>
BIOMD0000000006	<a href="http://identifiers.org/biomodels.db/BIOMD0000000006">http://identifiers.org/biomodels.db/BIOMD0000000006</a>
CL:0000007	<a href="http://identifiers.org/cl/CL:0000007">http://identifiers.org/cl/CL:0000007</a>
dc.creator	<a href="http://purl.org/dc/terms/creator">http://purl.org/dc/terms/creator</a>
dc.rights	<a href="http://purl.org/dc/terms/rights">http://purl.org/dc/terms/rights</a>
FP_001	<a href="http://www.obofoundry.org/wiki/index.php/FP_001_open">http://www.obofoundry.org/wiki/index.php/FP_001_open</a>
FP_002	<a href="http://www.obofoundry.org/wiki/index.php/FP_002_format">http://www.obofoundry.org/wiki/index.php/FP_002_format</a>
FP_003	<a href="http://www.obofoundry.org/wiki/index.php/FP_003_URLs">http://www.obofoundry.org/wiki/index.php/FP_003_URLs</a>
FP_004	<a href="http://www.obofoundry.org/wiki/index.php/FP_004_versioning">http://www.obofoundry.org/wiki/index.php/FP_004_versioning</a>
FP_005	<a href="http://www.obofoundry.org/wiki/index.php/FP_005_delineated_content">http://www.obofoundry.org/wiki/index.php/FP_005_delineated_content</a>
FP_006	<a href="http://www.obofoundry.org/wiki/index.php/FP_006_textual_definitions">http://www.obofoundry.org/wiki/index.php/FP_006_textual_definitions</a>
FP_007	<a href="http://www.obofoundry.org/wiki/index.php/FP_007_relations">http://www.obofoundry.org/wiki/index.php/FP_007_relations</a>
FP_008	<a href="http://www.obofoundry.org/wiki/index.php/FP_008_documented">http://www.obofoundry.org/wiki/index.php/FP_008_documented</a>
FP_010	<a href="http://www.obofoundry.org/wiki/index.php/FP_010_collaboration">http://www.obofoundry.org/wiki/index.php/FP_010_collaboration</a>
FP_012	<a href="http://www.obofoundry.org/wiki/index.php/FP_012_naming_conventions">http://www.obofoundry.org/wiki/index.php/FP_012_naming_conventions</a>
FP_013	<a href="http://www.obofoundry.org/wiki/index.php/FP_013_genus_differentia">http://www.obofoundry.org/wiki/index.php/FP_013_genus_differentia</a>
FP_014	<a href="http://www.obofoundry.org/wiki/index.php/FP_014_BF0">http://www.obofoundry.org/wiki/index.php/FP_014_BF0</a>
FP_018	<a href="http://www.obofoundry.org/wiki/index.php/FP_018_orthogonality">http://www.obofoundry.org/wiki/index.php/FP_018_orthogonality</a>
G0:0000079	<a href="http://identifiers.org/go/G0:0000079">http://identifiers.org/go/G0:0000079</a>
G0:0000278	<a href="http://identifiers.org/go/G0:0000278">http://identifiers.org/go/G0:0000278</a>
G0:0007050	<a href="http://identifiers.org/go/G0:0007050">http://identifiers.org/go/G0:0007050</a>
G0:0008150	<a href="http://identifiers.org/go/G0:0008150">http://identifiers.org/go/G0:0008150</a>

KISAO_0000019	<a href="http://identifiers.org/biomodels.kisao/KISAO_0000019">http://identifiers.org/biomodels.kisao/KISAO_0000019</a>
KISAO_0000088	<a href="http://identifiers.org/biomodels.kisao/KISAO_0000088">http://identifiers.org/biomodels.kisao/KISAO_0000088</a>
KISAO_0000097	<a href="http://identifiers.org/kisao/KISAO_0000097">http://identifiers.org/kisao/KISAO_0000097</a>
KISAO_0000280	<a href="http://identifiers.org/biomodels.kisao/KISAO_0000280">http://identifiers.org/biomodels.kisao/KISAO_0000280</a>
P04551	<a href="http://identifiers.org/uniprot/P04551">http://identifiers.org/uniprot/P04551</a>
PATO:0000033	<a href="http://identifiers.org/pato/PATO:0000033">http://identifiers.org/pato/PATO:0000033</a>
REACT_6308	<a href="http://identifiers.org/reactome/REACT_6308">http://identifiers.org/reactome/REACT_6308</a>
rel-2007-06-04	<a href="http://identifiers.org/combine.specifications/teddy.rel-2007-06-04">http://identifiers.org/combine.specifications/teddy.rel-2007-06-04</a>
rel-2014-04-24	<a href="http://identifiers.org/combine.specifications/teddy.rel-2014-04-24">http://identifiers.org/combine.specifications/teddy.rel-2014-04-24</a>
SB0:0000001	<a href="http://identifiers.org/biomodels.sbo/SB0:0000001">http://identifiers.org/biomodels.sbo/SB0:0000001</a>
SB0:0000004	<a href="http://identifiers.org/biomodels.sbo/SB0:0000004">http://identifiers.org/biomodels.sbo/SB0:0000004</a>
SB0:0000012	<a href="http://identifiers.org/biomodels.sbo/SB0:0000012">http://identifiers.org/biomodels.sbo/SB0:0000012</a>
SB0:0000180	<a href="http://identifiers.org/biomodels.sbo/SB0:0000180">http://identifiers.org/biomodels.sbo/SB0:0000180</a>
SB0:0000240	<a href="http://identifiers.org/biomodels.sbo/SB0:0000240">http://identifiers.org/biomodels.sbo/SB0:0000240</a>
SB0:0000252	<a href="http://identifiers.org/biomodels.sbo/SB0:0000252">http://identifiers.org/biomodels.sbo/SB0:0000252</a>
SB0:0000293	<a href="http://identifiers.org/biomodels.sbo/SB0:0000293">http://identifiers.org/biomodels.sbo/SB0:0000293</a>
SB0:0000472	<a href="http://identifiers.org/biomodels.sbo/SB0:0000472">http://identifiers.org/biomodels.sbo/SB0:0000472</a>
SB0:0000545	<a href="http://identifiers.org/biomodels.sbo/SB0:0000545">http://identifiers.org/biomodels.sbo/SB0:0000545</a>
TEDDY_0000000	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000000">http://identifiers.org/biomodels.teddy/TEDDY_0000000</a>
TEDDY_0000002	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000002">http://identifiers.org/biomodels.teddy/TEDDY_0000002</a>
TEDDY_0000003	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000003">http://identifiers.org/biomodels.teddy/TEDDY_0000003</a>
TEDDY_0000012	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000012">http://identifiers.org/biomodels.teddy/TEDDY_0000012</a>
TEDDY_0000015	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000015">http://identifiers.org/biomodels.teddy/TEDDY_0000015</a>
TEDDY_0000023	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000023">http://identifiers.org/biomodels.teddy/TEDDY_0000023</a>
TEDDY_0000025	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000025">http://identifiers.org/biomodels.teddy/TEDDY_0000025</a>
TEDDY_0000033	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000033">http://identifiers.org/biomodels.teddy/TEDDY_0000033</a>
TEDDY_0000034	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000034">http://identifiers.org/biomodels.teddy/TEDDY_0000034</a>
TEDDY_0000036	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000036">http://identifiers.org/biomodels.teddy/TEDDY_0000036</a>
TEDDY_0000050	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000050">http://identifiers.org/biomodels.teddy/TEDDY_0000050</a>
TEDDY_0000051	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000051">http://identifiers.org/biomodels.teddy/TEDDY_0000051</a>
TEDDY_0000053	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000053">http://identifiers.org/biomodels.teddy/TEDDY_0000053</a>
TEDDY_0000054	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000054">http://identifiers.org/biomodels.teddy/TEDDY_0000054</a>
TEDDY_0000055	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000055">http://identifiers.org/biomodels.teddy/TEDDY_0000055</a>
TEDDY_0000065	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000065">http://identifiers.org/biomodels.teddy/TEDDY_0000065</a>
TEDDY_0000067	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000067">http://identifiers.org/biomodels.teddy/TEDDY_0000067</a>
TEDDY_0000069	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000069">http://identifiers.org/biomodels.teddy/TEDDY_0000069</a>
TEDDY_0000072	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000072">http://identifiers.org/biomodels.teddy/TEDDY_0000072</a>
TEDDY_0000073	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000073">http://identifiers.org/biomodels.teddy/TEDDY_0000073</a>
TEDDY_0000074	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000074">http://identifiers.org/biomodels.teddy/TEDDY_0000074</a>
TEDDY_0000083	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000083">http://identifiers.org/biomodels.teddy/TEDDY_0000083</a>
TEDDY_0000085	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000085">http://identifiers.org/biomodels.teddy/TEDDY_0000085</a>
TEDDY_0000086	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000086">http://identifiers.org/biomodels.teddy/TEDDY_0000086</a>
TEDDY_0000088	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000088">http://identifiers.org/biomodels.teddy/TEDDY_0000088</a>
TEDDY_0000094	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000094">http://identifiers.org/biomodels.teddy/TEDDY_0000094</a>
TEDDY_0000099	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000099">http://identifiers.org/biomodels.teddy/TEDDY_0000099</a>
TEDDY_0000102	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000102">http://identifiers.org/biomodels.teddy/TEDDY_0000102</a>

TEDDY_0000106	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000106">http://identifiers.org/biomodels.teddy/TEDDY_0000106</a>
TEDDY_0000108	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000108">http://identifiers.org/biomodels.teddy/TEDDY_0000108</a>
TEDDY_0000109	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000109">http://identifiers.org/biomodels.teddy/TEDDY_0000109</a>
TEDDY_0000110	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000110">http://identifiers.org/biomodels.teddy/TEDDY_0000110</a>
TEDDY_0000111	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000111">http://identifiers.org/biomodels.teddy/TEDDY_0000111</a>
TEDDY_0000114	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000114">http://identifiers.org/biomodels.teddy/TEDDY_0000114</a>
TEDDY_0000116	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000116">http://identifiers.org/biomodels.teddy/TEDDY_0000116</a>
TEDDY_0000120	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000120">http://identifiers.org/biomodels.teddy/TEDDY_0000120</a>
TEDDY_0000126	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000126">http://identifiers.org/biomodels.teddy/TEDDY_0000126</a>
TEDDY_0000127	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000127">http://identifiers.org/biomodels.teddy/TEDDY_0000127</a>
TEDDY_0000128	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000128">http://identifiers.org/biomodels.teddy/TEDDY_0000128</a>
TEDDY_0000130	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000130">http://identifiers.org/biomodels.teddy/TEDDY_0000130</a>
TEDDY_0000131	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000131">http://identifiers.org/biomodels.teddy/TEDDY_0000131</a>
TEDDY_0000132	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000132">http://identifiers.org/biomodels.teddy/TEDDY_0000132</a>
TEDDY_0000143	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000143">http://identifiers.org/biomodels.teddy/TEDDY_0000143</a>
TEDDY_0000144	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000144">http://identifiers.org/biomodels.teddy/TEDDY_0000144</a>
TEDDY_0000146	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000146">http://identifiers.org/biomodels.teddy/TEDDY_0000146</a>
TEDDY_0000147	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000147">http://identifiers.org/biomodels.teddy/TEDDY_0000147</a>
TEDDY_0000148	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000148">http://identifiers.org/biomodels.teddy/TEDDY_0000148</a>
TEDDY_0000178	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000178">http://identifiers.org/biomodels.teddy/TEDDY_0000178</a>
TEDDY_0000179	<a href="http://identifiers.org/biomodels.teddy/TEDDY_0000179">http://identifiers.org/biomodels.teddy/TEDDY_0000179</a>
TEDDY_obsolete	<a href="http://identifiers.org/biomodels.teddy/TEDDY_obsolete">http://identifiers.org/biomodels.teddy/TEDDY_obsolete</a>
U0:0000061	<a href="http://identifiers.org/unit/U0:0000061">http://identifiers.org/unit/U0:0000061</a>



## C. SBML-Kodierung von Modell 1

Im folgenden ist die vollständige [SBML](#)-Kodierung (Level 2, Version 4) von Modell 1 gelistet. Umgebrochene Zeilen sind durch  $\hookrightarrow$  gekennzeichnet. Die Zeilennummern dienen der Referenzierung im Haupttext. Quelle: [BioModels Database](#) ([Le Novère et al. 2006](#)), Eintrag [BIOMD0000000005](#), [SBML](#)-Version abgerufen am 04.06.2014 von <http://www.ebi.ac.uk/biomodels-main/download?mid=BIOMD0000000005>.

```
1 <?xml version='1.0' encoding='UTF-8' standalone='no'?>
2
3 <!-- This model was downloaded from BioModels Database -->
4 <!-- Wed Jun 04 13:32:20 BST 2014 -->
5 <!-- http://www.ebi.ac.uk/biomodels/ -->
6
7 <sbml xmlns="http://www.sbml.org/sbml/level2/version4" level="2"
8     metaid="_000000" version="4">
9   <model id="BIOMD0000000005" name="Tyson1991 - Cell Cycle 6 var"
10     metaid="_000001">
11     <notes>
12       <body xmlns="http://www.w3.org/1999/xhtml">
13         <div class="dc:title">Tyson1991 - Cell Cycle 6 var</div>
14         <div class="dc:description">
15           <p>Mathematical model of the interactions of cdc2 and cyclin.</p>
16         </div>
17         <div class="dc:bibliographicCitation">
18           <p>This model is described in the article:</p>
19           <div class="bibo:title">
20             <a href="http://identifiers.org/pubmed/1831270" title="Access to this
21             ↪ publication">Modeling the cell division cycle: cdc2 and cyclin interactions.</a>
22           </div>
23           <div class="bibo:authorList">Tyson JJ.</div>
24           <div class="bibo:Journal">
25             Proc. Natl. Acad. Sci. U.S.A. 1991; 88(16); 7328-32
26           </div>
27           <p>Abstract:</p>
28           <div class="bibo:abstract">
29             <p>The proteins cdc2 and cyclin form a heterodimer
30             (maturation promoting factor) that controls the major
31             events of the cell cycle. A mathematical model for the
32             interactions of cdc2 and cyclin is constructed. Simulation
33             and analysis of the model show that the control system can
34             operate in three modes: as a steady state with high
35             maturation promoting factor activity, as a spontaneous
36             oscillator, or as an excitable switch. We associate the
37             steady state with metaphase arrest in unfertilized eggs,
38             the spontaneous oscillations with rapid division cycles in
39             early embryos, and the excitable switch with
40             growth-controlled division cycles typical of nonembryonic
```

```
40         cells.</p>
41     </div>
42 </div>
43     <div class="dc:publisher">
44         <p>This model is hosted on <a href="http://www.ebi.ac.uk/biomodels/">BioModels
↳ Database</a>
45         and identified by: <a href="http://identifiers.org/biomodels.db/BIOMD0000000005
↳ ">BIOMD0000000005</a>.
46     </p>
47     <p>To cite BioModels Database, please use:
48     <a href="http://identifiers.org/pubmed/20587024" title="Latest BioModels
↳ Database publication">BioModels Database: An enhanced, curated and annotated
↳ resource for published quantitative kinetic models</a>.
49 </p>
50 </div>
51 <div class="dc:license">
52     <p>To the extent possible under law, all copyright and
53     related or neighbouring rights to this encoded model have
54     been dedicated to the public domain worldwide. Please refer to
55     <a href="http://creativecommons.org/publicdomain/zero/1.0/" title="Access to:
↳ CC0 1.0 Universal (CC0 1.0), Public Domain Dedication">CC0 Public Domain
↳ Dedication</a> for more information.</p>
56 </div>
57 </body>
58 </notes>
59 <annotation>
60     <rdf:RDF
61         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
62         xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
63         xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#"
64         xmlns:dc="http://purl.org/dc/elements/1.1/"
65         xmlns:dcterms="http://purl.org/dc/terms/"
66         xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
67     <rdf:Description rdf:about="#_000001">
68         <dc:creator>
69             <rdf:Bag>
70                 <rdf:li rdf:parseType="Resource">
71                     <vCard:N rdf:parseType="Resource">
72                         <vCard:Family>Shapiro</vCard:Family>
73                         <vCard:Given>Bruce</vCard:Given>
74                     </vCard:N>
75                     <vCard:EMAIL>bshapiro@jpl.nasa.gov</vCard:EMAIL>
76                     <vCard:ORG rdf:parseType="Resource">
77                         <vCard:Orgname>
78                             NASA Jet Propulsion Laboratory
79                         </vCard:Orgname>
80                     </vCard:ORG>
81                 </rdf:li>
82                 <rdf:li rdf:parseType="Resource">
83                     <vCard:N rdf:parseType="Resource">
84                         <vCard:Family>Chelliah</vCard:Family>
85                         <vCard:Given>Vijayalakshmi</vCard:Given>
86                     </vCard:N>
87                     <vCard:EMAIL>viji@ebi.ac.uk</vCard:EMAIL>
88                     <vCard:ORG rdf:parseType="Resource">
89                         <vCard:Orgname>EMBL-EBI</vCard:Orgname>
```

---

```

90         </vCard:ORG>
91     </rdf:li>
92 </rdf:Bag>
93 </dc:creator>
94 <dcterms:created rdf:parseType="Resource">
95     <dcterms:W3CDTF>2005-02-08T18:28:27Z</dcterms:W3CDTF>
96 </dcterms:created>
97 <dcterms:modified rdf:parseType="Resource">
98     <dcterms:W3CDTF>2013-05-16T14:40:43Z</dcterms:W3CDTF>
99 </dcterms:modified>
100 <bqmodel:is>
101     <rdf:Bag>
102         <rdf:li rdf:resource="http://identifiers.org/biomodels.db/MODEL6614644188"/>
103     </rdf:Bag>
104 </bqmodel:is>
105 <bqmodel:is>
106     <rdf:Bag>
107         <rdf:li rdf:resource="http://identifiers.org/biomodels.db/BIOMD0000000005"/>
108     </rdf:Bag>
109 </bqmodel:is>
110 <bqmodel:isDescribedBy>
111     <rdf:Bag>
112         <rdf:li rdf:resource="http://identifiers.org/pubmed/1831270"/>
113     </rdf:Bag>
114 </bqmodel:isDescribedBy>
115 <bqbiol:occursIn>
116     <rdf:Bag>
117         <rdf:li rdf:resource="http://identifiers.org/taxonomy/33154"/>
118     </rdf:Bag>
119 </bqbiol:occursIn>
120 <bqbiol:isVersionOf>
121     <rdf:Bag>
122         <rdf:li rdf:resource="http://identifiers.org/kegg.pathway/sce04111"/>
123         <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0000278"/>
124     </rdf:Bag>
125 </bqbiol:isVersionOf>
126 <bqbiol:hasVersion>
127     <rdf:Bag>
128         <rdf:li rdf:resource="http://identifiers.org/reactome/REACT_152"/>
129     </rdf:Bag>
130 </bqbiol:hasVersion>
131 </rdf:Description>
132 </rdf:RDF>
133 </annotation>
134 <listOfCompartments>
135     <compartment id="cell" metaid="_000002" size="1">
136         <annotation>
137             <rdf:RDF
138                 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
139                 xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
140                 xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
141                 <rdf:Description rdf:about="#_000002">
142                     <bqbiol:is>
143                         <rdf:Bag>
144                             <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0005623"/>
145                         </rdf:Bag>

```



```
146         </bqbiol:is>
147     </rdf:Description>
148 </rdf:RDF>
149 </annotation>
150 </compartment>
151 </listOfCompartments>
152 <listOfSpecies>
153     <species id="EmptySet" initialAmount="0" metaid="_000003" boundaryCondition="true"
154     ↪ compartment="cell"/>
155     <species id="C2" initialAmount="0" name="cdc2k" metaid="_000004" compartment="cell">
156         <annotation>
157             <rdf:RDF
158                 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
159                 xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
160                 xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
161                 <rdf:Description rdf:about="#_000004">
162                     <bqbiol:isVersionOf>
163                         <rdf:Bag>
164                             <rdf:li rdf:resource="http://identifiers.org/uniprot/P04551"/>
165                         </rdf:Bag>
166                     </bqbiol:isVersionOf>
167                 </rdf:Description>
168             </rdf:RDF>
169         </annotation>
170     </species>
171     <species id="CP" initialAmount="0.75" name="cdc2k-P" metaid="_000005" compartment="
172     ↪ cell">
173         <annotation>
174             <rdf:RDF
175                 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
176                 xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
177                 xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
178                 <rdf:Description rdf:about="#_000005">
179                     <bqbiol:isVersionOf>
180                         <rdf:Bag>
181                             <rdf:li rdf:resource="http://identifiers.org/uniprot/P04551"/>
182                         </rdf:Bag>
183                     </bqbiol:isVersionOf>
184                 </rdf:Description>
185             </rdf:RDF>
186         </annotation>
187     </species>
188     <species id="M" initialAmount="0" name="p-cyclin_cdc2" metaid="_000006" compartment
189     ↪ ="cell">
190         <annotation>
191             <rdf:RDF
192                 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
193                 xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
194                 xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
195                 <rdf:Description rdf:about="#_000006">
196                     <bqbiol:hasPart>
197                         <rdf:Bag>
198                             <rdf:li rdf:resource="http://identifiers.org/uniprot/P04551"/>
199                             <rdf:li rdf:resource="http://identifiers.org/interpro/IPR006670"/>
200                         </rdf:Bag>
201                     </bqbiol:hasPart>
```

---

```

199         </rdf:Description>
200     </rdf:RDF>
201 </annotation>
202 </species>
203 <species id="pM" initialAmount="0.25" name="p-cyclin_cdc2-p" metaid="_000007"
↪ compartment="cell">
204     <annotation>
205         <rdf:RDF
206             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
207             xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
208             xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
209             <rdf:Description rdf:about="#_000007">
210                 <bqbiol:hasPart>
211                     <rdf:Bag>
212                         <rdf:li rdf:resource="http://identifiers.org/uniprot/P04551"/>
213                         <rdf:li rdf:resource="http://identifiers.org/interpro/IPR006670"/>
214                     </rdf:Bag>
215                 </bqbiol:hasPart>
216             </rdf:Description>
217         </rdf:RDF>
218     </annotation>
219 </species>
220 <species id="Y" initialAmount="0" name="cyclin" metaid="_000008" compartment="cell">
221     <annotation>
222         <rdf:RDF
223             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
224             xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
225             xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
226             <rdf:Description rdf:about="#_000008">
227                 <bqbiol:isVersionOf>
228                     <rdf:Bag>
229                         <rdf:li rdf:resource="http://identifiers.org/interpro/IPR006670"/>
230                     </rdf:Bag>
231                 </bqbiol:isVersionOf>
232             </rdf:Description>
233         </rdf:RDF>
234     </annotation>
235 </species>
236 <species id="YP" initialAmount="0" name="p-cyclin" metaid="_000009" compartment="
↪ cell">
237     <annotation>
238         <rdf:RDF
239             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
240             xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
241             xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
242             <rdf:Description rdf:about="#_000009">
243                 <bqbiol:isVersionOf>
244                     <rdf:Bag>
245                         <rdf:li rdf:resource="http://identifiers.org/interpro/IPR006670"/>
246                     </rdf:Bag>
247                 </bqbiol:isVersionOf>
248             </rdf:Description>
249         </rdf:RDF>
250     </annotation>
251 </species>

```

```
252     <species id="YT" name="total_cyclin" metaid="metaid_0000020" boundaryCondition="true
↪ " compartment="cell">
253       <annotation>
254         <rdf:RDF
255           xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
256           xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
257           xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
258           <rdf:Description rdf:about="#metaid_0000020">
259             <bqbiol:isVersionOf>
260               <rdf:Bag>
261                 <rdf:li rdf:resource="http://identifiers.org/interpro/IPR006670"/>
262               </rdf:Bag>
263             </bqbiol:isVersionOf>
264           </rdf:Description>
265         </rdf:RDF>
266       </annotation>
267     </species>
268     <species id="CT" name="total_cdc2" metaid="metaid_0000022" boundaryCondition="true"
↪ compartment="cell">
269       <annotation>
270         <rdf:RDF
271           xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
272           xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
273           xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
274           <rdf:Description rdf:about="#metaid_0000022">
275             <bqbiol:isVersionOf>
276               <rdf:Bag>
277                 <rdf:li rdf:resource="http://identifiers.org/uniprot/P04551"/>
278               </rdf:Bag>
279             </bqbiol:isVersionOf>
280           </rdf:Description>
281         </rdf:RDF>
282       </annotation>
283     </species>
284 </listOfSpecies>
285 <listOfRules>
286   <assignmentRule metaid="metaid_0000021" variable="YT">
287     <math xmlns="http://www.w3.org/1998/Math/MathML">
288       <apply>
289         <plus/>
290         <ci> Y </ci>
291         <ci> YP </ci>
292         <ci> M </ci>
293         <ci> pM </ci>
294       </apply>
295     </math>
296   </assignmentRule>
297   <assignmentRule metaid="metaid_0000023" variable="CT">
298     <math xmlns="http://www.w3.org/1998/Math/MathML">
299       <apply>
300         <plus/>
301         <ci> C2 </ci>
302         <ci> CP </ci>
303         <ci> M </ci>
304         <ci> pM </ci>
305       </apply>
```

---

```

306     </math>
307   </assignmentRule>
308 </listOfRules>
309 <listOfReactions>
310   <reaction id="Reaction1" name="cyclin_cdc2k dissociation" metaid="_000010"
    ↪ reversible="false">
311     <annotation>
312       <rdf:RDF
313         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
314         xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
315         xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
316       <rdf:Description rdf:about="#_000010">
317         <bqbiol:hasVersion>
318           <rdf:Bag>
319             <rdf:li rdf:resource="http://identifiers.org/reactome/REACT_6308"/>
320           </rdf:Bag>
321         </bqbiol:hasVersion>
322         <bqbiol:isVersionOf>
323           <rdf:Bag>
324             <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0000079"/>
325           </rdf:Bag>
326         </bqbiol:isVersionOf>
327       </rdf:Description>
328     </rdf:RDF>
329   </annotation>
330   <listOfReactants>
331     <speciesReference species="M" metaid="_712369"/>
332   </listOfReactants>
333   <listOfProducts>
334     <speciesReference species="C2" metaid="_712382"/>
335     <speciesReference species="YP" metaid="_712394"/>
336   </listOfProducts>
337   <kineticLaw metaid="_712406">
338     <math xmlns="http://www.w3.org/1998/Math/MathML">
339       <apply>
340         <times/>
341         <ci> cell </ci>
342         <ci> k6 </ci>
343         <ci> M </ci>
344       </apply>
345     </math>
346     <listOfParameters>
347       <parameter id="k6" metaid="_840210" value="1"/>
348     </listOfParameters>
349   </kineticLaw>
350 </reaction>
351 <reaction id="Reaction2" name="cdc2k phosphorylation" metaid="_000011" reversible="
    ↪ false">
352   <annotation>
353     <rdf:RDF
354       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
355       xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
356       xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
357     <rdf:Description rdf:about="#_000011">
358       <bqbiol:isVersionOf>
359         <rdf:Bag>

```

```
360         <rdf:li rdf:resource="http://identifiers.org/ec-code/2.7.11.1"/>
361         <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0006468"/>
362     </rdf:Bag>
363     </bqbiol:isVersionOf>
364 </rdf:Description>
365 </rdf:RDF>
366 </annotation>
367 <listOfReactants>
368     <speciesReference species="C2" metaid="_712418"/>
369 </listOfReactants>
370 <listOfProducts>
371     <speciesReference species="CP" metaid="_712430"/>
372 </listOfProducts>
373 <kineticLaw metaid="_712442">
374     <math xmlns="http://www.w3.org/1998/Math/MathML">
375         <apply>
376             <times/>
377             <ci> cell </ci>
378             <ci> C2 </ci>
379             <ci> k8notP </ci>
380         </apply>
381     </math>
382     <listOfParameters>
383         <parameter id="k8notP" metaid="_840213" value="1000000"/>
384     </listOfParameters>
385 </kineticLaw>
386 </reaction>
387 <reaction id="Reaction3" name="cdc2k dephosphorylation" metaid="_000012" reversible
↪ ="false">
388     <annotation>
389         <rdf:RDF
390             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
391             xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
392             xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
393         <rdf:Description rdf:about="#_000012">
394             <bqbiol:isVersionOf>
395                 <rdf:Bag>
396                     <rdf:li rdf:resource="http://identifiers.org/ec-code/3.1.3.16"/>
397                     <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0006470"/>
398                 </rdf:Bag>
399             </bqbiol:isVersionOf>
400         </rdf:Description>
401     </rdf:RDF>
402 </annotation>
403 <listOfReactants>
404     <speciesReference species="CP" metaid="_712454"/>
405 </listOfReactants>
406 <listOfProducts>
407     <speciesReference species="C2" metaid="_712466"/>
408 </listOfProducts>
409 <kineticLaw metaid="_712478">
410     <math xmlns="http://www.w3.org/1998/Math/MathML">
411         <apply>
412             <times/>
413             <ci> cell </ci>
414             <ci> CP </ci>
```

---

```

415         <ci> k9 </ci>
416     </apply>
417 </math>
418 <listOfParameters>
419     <parameter id="k9" metaid="_840215" value="1000"/>
420 </listOfParameters>
421 </kineticLaw>
422 </reaction>
423 <reaction id="Reaction4" name="cyclin cdc2k-p association" metaid="_000013"
↪ reversible="false">
424     <annotation>
425         <rdf:RDF
426             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
427             xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
428             xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
429         <rdf:Description rdf:about="#_000013">
430             <bqbiol:hasVersion>
431                 <rdf:Bag>
432                     <rdf:li rdf:resource="http://identifiers.org/reactome/REACT_6216"/>
433                     <rdf:li rdf:resource="http://identifiers.org/reactome/REACT_6308"/>
434                 </rdf:Bag>
435             </bqbiol:hasVersion>
436         </rdf:Description>
437     </rdf:RDF>
438 </annotation>
439 <listOfReactants>
440     <speciesReference species="CP" metaid="_712490"/>
441     <speciesReference species="Y" metaid="_712502"/>
442 </listOfReactants>
443 <listOfProducts>
444     <speciesReference species="pM" metaid="_712514"/>
445 </listOfProducts>
446 <kineticLaw metaid="_712526">
447     <math xmlns="http://www.w3.org/1998/Math/MathML">
448         <apply>
449             <times/>
450             <ci> cell </ci>
451             <ci> CP </ci>
452             <ci> k3 </ci>
453             <ci> Y </ci>
454         </apply>
455     </math>
456     <listOfParameters>
457         <parameter id="k3" metaid="_840217" value="200"/>
458     </listOfParameters>
459 </kineticLaw>
460 </reaction>
461 <reaction id="Reaction5" name="deactivation of cdc2 kinase" metaid="_000014"
↪ reversible="false">
462     <annotation>
463         <rdf:RDF
464             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
465             xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
466             xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
467         <rdf:Description rdf:about="#_000014">
468             <bqbiol:isVersionOf>

```

```
469         <rdf:Bag>
470           <rdf:li rdf:resource="http://identifiers.org/ec-code/2.7.10.2"/>
471           <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0006468"/>
472           <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0045736"/>
473         </rdf:Bag>
474       </bqbiol:isVersionOf>
475       <bqbiol:hasVersion>
476         <rdf:Bag>
477           <rdf:li rdf:resource="http://identifiers.org/reactome/REACT_3178"/>
478           <rdf:li rdf:resource="http://identifiers.org/reactome/REACT_6327"/>
479         </rdf:Bag>
480       </bqbiol:hasVersion>
481     </rdf:Description>
482   </rdf:RDF>
483 </annotation>
484 <listOfReactants>
485   <speciesReference species="M" metaid="_712538"/>
486 </listOfReactants>
487 <listOfProducts>
488   <speciesReference species="pM" metaid="_712550"/>
489 </listOfProducts>
490 <kineticLaw metaid="_712562">
491   <math xmlns="http://www.w3.org/1998/Math/MathML">
492     <apply>
493       <times/>
494       <ci> cell </ci>
495       <ci> k5notP </ci>
496       <ci> M </ci>
497     </apply>
498   </math>
499   <listOfParameters>
500     <parameter id="k5notP" metaid="_840219" value="0"/>
501   </listOfParameters>
502 </kineticLaw>
503 </reaction>
504 <reaction id="Reaction6" name="cyclin biosynthesis" metaid="_000015" reversible="
↪ false">
505   <annotation>
506     <rdf:RDF
507       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
508       xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
509       xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
510       <rdf:Description rdf:about="#_000015">
511         <bqbiol:isVersionOf>
512           <rdf:Bag>
513             <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0006412"/>
514           </rdf:Bag>
515         </bqbiol:isVersionOf>
516       </rdf:Description>
517     </rdf:RDF>
518   </annotation>
519   <listOfReactants>
520     <speciesReference species="EmptySet" metaid="_712574"/>
521   </listOfReactants>
522   <listOfProducts>
523     <speciesReference species="Y" metaid="_712586"/>
```



---

```

524     </listOfProducts>
525     <kineticLaw metaid="_712598">
526       <math xmlns="http://www.w3.org/1998/Math/MathML">
527         <apply>
528           <times/>
529           <ci> cell </ci>
530           <ci> k1aa </ci>
531         </apply>
532       </math>
533       <listOfParameters>
534         <parameter id="k1aa" metaid="_840221" value="0.015"/>
535       </listOfParameters>
536     </kineticLaw>
537   </reaction>
538   <reaction id="Reaction7" name="default degradation of cyclin" metaid="_000016"
↪ reversible="false">
539     <annotation>
540       <rdf:RDF
541         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
542         xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
543         xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
544       <rdf:Description rdf:about="#_000016">
545         <bqbiol:isVersionOf>
546           <rdf:Bag>
547             <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0008054"/>
548           </rdf:Bag>
549         </bqbiol:isVersionOf>
550       </rdf:Description>
551     </rdf:RDF>
552   </annotation>
553   <listOfReactants>
554     <speciesReference species="Y" metaid="_712610"/>
555   </listOfReactants>
556   <listOfProducts>
557     <speciesReference species="EmptySet" metaid="_712622"/>
558   </listOfProducts>
559   <kineticLaw metaid="_712634">
560     <math xmlns="http://www.w3.org/1998/Math/MathML">
561       <apply>
562         <times/>
563         <ci> cell </ci>
564         <ci> k2 </ci>
565         <ci> Y </ci>
566       </apply>
567     </math>
568     <listOfParameters>
569       <parameter id="k2" metaid="_840223" value="0"/>
570     </listOfParameters>
571   </kineticLaw>
572 </reaction>
573 <reaction id="Reaction8" name="cdc2 kinase triggered degration of cyclin" metaid="
↪ _000017" reversible="false">
574   <annotation>
575     <rdf:RDF
576       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
577       xmlns:bqmodel="http://biomodels.net/model-qualifiers/"

```

```
578         xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
579         <rdf:Description rdf:about="#_000017">
580             <bqbiol:isVersionOf>
581                 <rdf:Bag>
582                     <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0008054"/>
583                 </rdf:Bag>
584             </bqbiol:isVersionOf>
585         </rdf:Description>
586     </rdf:RDF>
587 </annotation>
588 <listOfReactants>
589     <speciesReference species="YP" metaid="_712646"/>
590 </listOfReactants>
591 <listOfProducts>
592     <speciesReference species="EmptySet" metaid="_712658"/>
593 </listOfProducts>
594 <kineticLaw metaid="_712670">
595     <math xmlns="http://www.w3.org/1998/Math/MathML">
596         <apply>
597             <times/>
598             <ci> cell </ci>
599             <ci> k7 </ci>
600             <ci> YP </ci>
601         </apply>
602     </math>
603     <listOfParameters>
604         <parameter id="k7" metaid="_840225" value="0.6"/>
605     </listOfParameters>
606 </kineticLaw>
607 </reaction>
608 <reaction id="Reaction9" name="activation of cdc2 kinase" metaid="_000018"
↪ reversible="false">
609     <annotation>
610         <rdf:RDF
611             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
612             xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
613             xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
614             <rdf:Description rdf:about="#_000018">
615                 <bqbiol:hasVersion>
616                     <rdf:Bag>
617                         <rdf:li rdf:resource="http://identifiers.org/reactome/REACT_6257"/>
618                         <rdf:li rdf:resource="http://identifiers.org/reactome/REACT_6175"/>
619                         <rdf:li rdf:resource="http://identifiers.org/reactome/REACT_6294"/>
620                     </rdf:Bag>
621                 </bqbiol:hasVersion>
622                 <bqbiol:isVersionOf>
623                     <rdf:Bag>
624                         <rdf:li rdf:resource="http://identifiers.org/ec-code/3.1.3.16"/>
625                         <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0006470"/>
626                         <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0045737"/>
627                     </rdf:Bag>
628                 </bqbiol:isVersionOf>
629             </rdf:Description>
630         </rdf:RDF>
631     </annotation>
632     <listOfReactants>
```

---

```

633     <speciesReference species="pM" metaid="_712682"/>
634 </listOfReactants>
635 <listOfProducts>
636     <speciesReference species="M" metaid="_712694"/>
637 </listOfProducts>
638 <listOfModifiers>
639     <modifierSpeciesReference species="CT" metaid="_712706"/>
640 </listOfModifiers>
641 <kineticLaw metaid="_712718">
642     <math xmlns="http://www.w3.org/1998/Math/MathML">
643         <apply>
644             <times/>
645             <ci> cell </ci>
646             <ci> pM </ci>
647             <apply>
648                 <plus/>
649                 <ci> k4prime </ci>
650                 <apply>
651                     <times/>
652                     <ci> k4 </ci>
653                     <apply>
654                         <power/>
655                         <apply>
656                             <divide/>
657                             <ci> M </ci>
658                             <ci> CT </ci>
659                         </apply>
660                         <cn type="integer"> 2 </cn>
661                     </apply>
662                 </apply>
663             </apply>
664         </math>
665     <listOfParameters>
666         <parameter id="k4" metaid="_840227" value="180"/>
667         <parameter id="k4prime" metaid="_840229" value="0.018"/>
668     </listOfParameters>
669 </kineticLaw>
670 </reaction>
671 </listOfReactions>
672 </model>
673 </sbml>

```



## D. SBML-Kodierung von Modell 2

Im folgenden ist die vollständige [SBML](#)-Kodierung (Level 2, Version 4) von Modell 2 gelistet. Umgebrochene Zeilen sind durch  $\hookrightarrow$  gekennzeichnet. Die Zeilennummern dienen der Referenzierung im Haupttext. Quelle: [BioModels Database](#) ([Le NOVÈRE ET AL. 2006](#)), Eintrag [BIOMD0000000006](#), [SBML](#)-Version abgerufen am 04.06.2014 von <http://www.ebi.ac.uk/biomodels-main/download?mid=BIOMD0000000006>.

```
1 <?xml version='1.0' encoding='UTF-8' standalone='no'?>
2
3 <!-- This model was downloaded from BioModels Database -->
4 <!-- Wed Jun 04 13:34:30 BST 2014 -->
5 <!-- http://www.ebi.ac.uk/biomodels/ -->
6
7 <sbml xmlns="http://www.sbml.org/sbml/level2/version4" level="2"
8     metaid="_000000" version="4">
9   <model id="BIOMD0000000006" name="Tyson1991 - Cell Cycle 2 var"
10     metaid="_000001">
11     <notes>
12       <body xmlns="http://www.w3.org/1999/xhtml">
13         <div class="dc:title">Tyson1991 - Cell Cycle 2 var</div>
14         <div class="dc:description">
15           <p>Mathematical model of the interactions of cdc2 and cyclin.</p>
16         </div>
17         <div class="dc:provenance">
18           <p>Description taken from the original Cellerator version of
19             the model (<a href="http://www.cellerator.org/notebooks/Tyson2.html">Tyson
20             ↪ (1991, 2 variables)</a>
21             at <a href="http://www.cellerator.org/" title="xCellerator: A Mathematica
22             ↪ package for biological modeling">http://www.cellerator.org</a>).</p>
23         </div>
24         <div class="dc:bibliographicCitation">
25           <p>This model is described in the article:</p>
26           <div class="bibo:title">
27             <a href="http://identifiers.org/pubmed/1831270"
28               title="Access to this publication">Modeling the cell
29               division cycle: cdc2 and cyclin interactions.</a>
30           </div>
31           <div class="bibo:authorList">Tyson JJ.</div>
32           <div class="bibo:Journal">
33             Proc. Natl. Acad. Sci. U.S.A. 1991; 88(16); 7328-32
34           </div>
35           <p>Abstract:</p>
36           <div class="bibo:abstract">
37             <p>The proteins cdc2 and cyclin form a heterodimer
38               (maturation promoting factor) that controls the major
39               events of the cell cycle. A mathematical model for the
40               interactions of cdc2 and cyclin is constructed. Simulation
```

```

39         and analysis of the model show that the control system can
40         operate in three modes: as a steady state with high
41         maturation promoting factor activity, as a spontaneous
42         oscillator, or as an excitable switch. We associate the
43         steady state with metaphase arrest in unfertilized eggs,
44         the spontaneous oscillations with rapid division cycles in
45         early embryos, and the excitable switch with
46         growth-controlled division cycles typical of nonembryonic
47         cells.</p>
48     </div>
49 </div>
50 <div class="bm:modification">
51     <p>This is a two variable reduction of the larger 6-variable
52     model published in the same paper. The equations are:</p>
53     <p>u' = k4(v-u)(alpha+u^2)-k6*u <br/>
54     v'=kappa-k6*u <br/>
55     z= v-u <br/>
56     with kappa = k1[aa]/[CT]</p>
57     <p>In the present implementation, an additional variable z
58     is introduced with z = v-u is made, so that the different
59     variables be interpreted as follows:</p>
60     <p>u=[activeMPF]/[CT] <br/>
61     v=( [cyclin]+[preMPF]+[activeMPF])/[CT] <br/>
62     z=( [ cyclin]+[preMPF])/[CT] <br/>
63     with [CT]=[CDC2]+[CDC2P]+[preMPF]+[aMPF].</p>
64     <p>The reactions included are only to show the flows between
65     z and u, and do not influence the species, as they all are
66     set to <em>boundaryCondition=True</em>, meaning, that they
67     are only determined by the rate rules (explicit differential
68     equations) and assignment rules.</p>
69     <p>If you set <em>boundaryCondition=False</em> and remove
70     the rate rules for v, u and the the assignment rule for z,
71     you get the more symmetrical, but equivalent, version from
72     the Cellerator repository:</p>
73     <p>u' = k4*z*(alpha+u^2)-k6*u <br/>
74     z'=kappa-z*(alpha+u^2)</p>
75 </div>
76 <div class="dc:publisher">
77     <p>This model is hosted on <a href="http://www.ebi.ac.uk/biomodels/">BioModels
↪ Database</a>
78     and identified by: <a href="http://identifiers.org/biomodels.db/BIOMD0000000006
↪ ">BIOMD0000000006</a>.</p>
79     <p>To cite BioModels Database, please use:
80     <a href="http://identifiers.org/pubmed/20587024"
81     title="Latest BioModels Database publication">
82     BioModels Database: An enhanced, curated and annotated
83     resource for published quantitative kinetic
84     models</a>.</p>
85 </div>
86 <div class="dc:license">
87     <p>To the extent possible under law, all copyright and
88     related or neighbouring rights to this encoded model have
89     been dedicated to the public domain worldwide. Please refer
90     to <a href="http://creativecommons.org/publicdomain/zero/1.0/" title="Access to:
↪ CC0 1.0 Universal (CC0 1.0), Public Domain Dedication">CC0 Public Domain
↪ Dedication</a>

```

---

```

91         for more information.</p>
92     </div>
93 </body>
94 </notes>
95 <annotation>
96     <rdf:RDF
97         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
98         xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
99         xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#"
100        xmlns:dc="http://purl.org/dc/elements/1.1/"
101        xmlns:dcterms="http://purl.org/dc/terms/"
102        xmlns:bqbiol="http://biomodels.net/biology-qualifiers/"
103    <rdf:Description rdf:about="#_000001">
104        <dc:creator>
105            <rdf:Bag>
106                <rdf:li rdf:parseType="Resource">
107                    <vCard:N rdf:parseType="Resource">
108                        <vCard:Family>Shapiro</vCard:Family>
109                        <vCard:Given>Bruce</vCard:Given>
110                    </vCard:N>
111                    <vCard:EMAIL>bshapiro@jpl.nasa.gov</vCard:EMAIL>
112                    <vCard:ORG rdf:parseType="Resource">
113                        <vCard:Orgname>
114                            NASA Jet Propulsion Laboratory
115                        </vCard:Orgname>
116                    </vCard:ORG>
117                </rdf:li>
118                <rdf:li rdf:parseType="Resource">
119                    <vCard:N rdf:parseType="Resource">
120                        <vCard:Family>Endler</vCard:Family>
121                        <vCard:Given>Lukas</vCard:Given>
122                    </vCard:N>
123                    <vCard:EMAIL>lukas@ebi.ac.uk</vCard:EMAIL>
124                    <vCard:ORG rdf:parseType="Resource">
125                        <vCard:Orgname>EMBL-EBI</vCard:Orgname>
126                    </vCard:ORG>
127                </rdf:li>
128            </rdf:Bag>
129        </dc:creator>
130        <dcterms:created rdf:parseType="Resource">
131            <dcterms:W3CDTF>2005-02-08T18:36:17Z</dcterms:W3CDTF>
132        </dcterms:created>
133        <dcterms:modified rdf:parseType="Resource">
134            <dcterms:W3CDTF>2013-05-16T14:38:56Z</dcterms:W3CDTF>
135        </dcterms:modified>
136        <bqmodel:is>
137            <rdf:Bag>
138                <rdf:li rdf:resource="http://identifiers.org/biomodels.db/MODEL6614715255"/>
139            </rdf:Bag>
140        </bqmodel:is>
141        <bqmodel:is>
142            <rdf:Bag>
143                <rdf:li rdf:resource="http://identifiers.org/biomodels.db/BIOMD0000000006"/>
144            </rdf:Bag>
145        </bqmodel:is>
146        <bqmodel:isDescribedBy>

```



```
147         <rdf:Bag>
148           <rdf:li rdf:resource="http://identifiers.org/pubmed/1831270"/>
149         </rdf:Bag>
150       </bqmodel:isDescribedBy>
151     <bqbiol:occursIn>
152       <rdf:Bag>
153         <rdf:li rdf:resource="http://identifiers.org/taxonomy/33154"/>
154       </rdf:Bag>
155     </bqbiol:occursIn>
156     <bqbiol:is>
157       <rdf:Bag>
158         <rdf:li rdf:resource="http://identifiers.org/kegg.pathway/sce04111"/>
159       </rdf:Bag>
160     </bqbiol:is>
161     <bqbiol:isVersionOf>
162       <rdf:Bag>
163         <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0000278"/>
164       </rdf:Bag>
165     </bqbiol:isVersionOf>
166     <bqbiol:hasVersion>
167       <rdf:Bag>
168         <rdf:li rdf:resource="http://identifiers.org/reactome/REACT_152"/>
169       </rdf:Bag>
170     </bqbiol:hasVersion>
171   </rdf:Description>
172 </rdf:RDF>
173 </annotation>
174 <listOfCompartments>
175   <compartment id="cell" metaid="_000002" size="1">
176     <annotation>
177       <rdf:RDF
178         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
179         xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
180         xmlns:bqbiol="http://biomodels.net/biology-qualifiers/"
181         <rdf:Description rdf:about="#_000002">
182           <bqbiol:is>
183             <rdf:Bag>
184               <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0005623"/>
185             </rdf:Bag>
186           </bqbiol:is>
187         </rdf:Description>
188       </rdf:RDF>
189     </annotation>
190   </compartment>
191 </listOfCompartments>
192 <listOfSpecies>
193   <species id="EmptySet" initialAmount="1" hasOnlySubstanceUnits="true" metaid="
194   ↪ _808249" boundaryCondition="true" sboTerm="SBO:0000291" compartment="cell"/>
195   <species id="u" initialAmount="0" hasOnlySubstanceUnits="true" metaid="_808265"
196   ↪ boundaryCondition="true" sboTerm="SBO:0000297" compartment="cell">
197     <annotation>
198       <rdf:RDF
199         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
200         xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
201         xmlns:bqbiol="http://biomodels.net/biology-qualifiers/"
202         <rdf:Description rdf:about="#_808265">
```

---

```

201         <bqbiol:hasProperty>
202             <rdf:Bag>
203                 <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0004693"/>
204             </rdf:Bag>
205         </bqbiol:hasProperty>
206         <bqbiol:isVersionOf>
207             <rdf:Bag>
208                 <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0031387"/>
209             </rdf:Bag>
210         </bqbiol:isVersionOf>
211     </rdf:Description>
212 </rdf:RDF>
213 </annotation>
214 </species>
215 <species id="z" initialAmount="0" hasOnlySubstanceUnits="true" metaid="_808285"
    ⇨ boundaryCondition="true" sboTerm="SBO:0000297" compartment="cell">
216     <annotation>
217         <rdf:RDF
218             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
219             xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
220             xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
221             <rdf:Description rdf:about="#_808285">
222                 <bqbiol:hasVersion>
223                     <rdf:Bag>
224                         <rdf:li rdf:resource="http://identifiers.org/uniprot/P04551"/>
225                     </rdf:Bag>
226                 </bqbiol:hasVersion>
227             </rdf:Description>
228         </rdf:RDF>
229     </annotation>
230 </species>
231 <species id="v" initialAmount="0" hasOnlySubstanceUnits="true" metaid="_175119"
    ⇨ boundaryCondition="true" sboTerm="SBO:0000297" compartment="cell">
232     <annotation>
233         <rdf:RDF
234             xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
235             xmlns:bqmodel="http://biomodels.net/model-qualifiers/"
236             xmlns:bqbiol="http://biomodels.net/biology-qualifiers/">
237             <rdf:Description rdf:about="#_175119">
238                 <bqbiol:hasVersion>
239                     <rdf:Bag>
240                         <rdf:li rdf:resource="http://identifiers.org/uniprot/P04551"/>
241                     </rdf:Bag>
242                 </bqbiol:hasVersion>
243                 <bqbiol:hasVersion>
244                     <rdf:Bag>
245                         <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0031387"/>
246                     </rdf:Bag>
247                 </bqbiol:hasVersion>
248                 <bqbiol:hasProperty>
249                     <rdf:Bag>
250                         <rdf:li rdf:resource="http://identifiers.org/obo.go/G0:0004693"/>
251                     </rdf:Bag>
252                 </bqbiol:hasProperty>
253             </rdf:Description>
254         </rdf:RDF>

```

```
255     </annotation>
256   </species>
257 </listOfSpecies>
258 <listOfParameters>
259   <parameter id="kappa" metaid="_175134" value="0.015"/>
260   <parameter id="k6" metaid="_175135" value="1"/>
261   <parameter id="k4" metaid="_175136" value="180"/>
262   <parameter id="k4prime" metaid="_175139" value="0.018"/>
263   <parameter id="alpha" constant="false" metaid="_175140"/>
264 </listOfParameters>
265 <listOfRules>
266   <rateRule metaid="_175128" variable="u">
267     <math xmlns="http://www.w3.org/1998/Math/MathML">
268       <apply>
269         <minus/>
270         <apply>
271           <times/>
272           <ci> k4 </ci>
273           <apply>
274             <minus/>
275             <ci> v </ci>
276             <ci> u </ci>
277           </apply>
278         <apply>
279           <plus/>
280           <ci> alpha </ci>
281         <apply>
282           <power/>
283           <ci> u </ci>
284           <cn type="integer"> 2 </cn>
285         </apply>
286       </apply>
287     </math>
288   </rateRule>
289   <assignmentRule metaid="_175129" variable="z">
290     <math xmlns="http://www.w3.org/1998/Math/MathML">
291       <apply>
292         <minus/>
293         <ci> v </ci>
294         <ci> u </ci>
295       </apply>
296     </math>
297   </assignmentRule>
298   <rateRule metaid="_175131" variable="v">
299     <math xmlns="http://www.w3.org/1998/Math/MathML">
300       <apply>
301         <minus/>
302         <ci> kappa </ci>
303       </apply>
304     </math>
305   </rateRule>
306 </listOfRules>
307 </model>
308 </sbml>
```

---

```

311         <times/>
312         <ci> k6 </ci>
313         <ci> u </ci>
314     </apply>
315 </math>
316 </rateRule>
317 <assignmentRule metaid="_175132" variable="alpha">
318     <math xmlns="http://www.w3.org/1998/Math/MathML">
319         <apply>
320             <divide/>
321             <ci> k4prime </ci>
322             <ci> k4 </ci>
323         </apply>
324     </math>
325 </assignmentRule>
326 </listOfRules>
327 <listOfReactions>
328     <reaction id="Reaction1" metaid="_175121" reversible="false" sboTerm="SBO:0000205">
329         <listOfReactants>
330             <speciesReference species="EmptySet" metaid="_372744"/>
331         </listOfReactants>
332         <listOfProducts>
333             <speciesReference species="z" metaid="_372756"/>
334         </listOfProducts>
335         <kineticLaw metaid="_372768">
336             <math xmlns="http://www.w3.org/1998/Math/MathML">
337                 <ci> kappa </ci>
338             </math>
339         </kineticLaw>
340     </reaction>
341     <reaction id="Reaction2" metaid="_175122" reversible="false" sboTerm="SBO:0000179">
342         <listOfReactants>
343             <speciesReference species="u" metaid="_372780"/>
344         </listOfReactants>
345         <listOfProducts>
346             <speciesReference species="EmptySet" metaid="_372792"/>
347         </listOfProducts>
348         <kineticLaw metaid="_372804">
349             <math xmlns="http://www.w3.org/1998/Math/MathML">
350                 <apply>
351                     <times/>
352                     <ci> k6 </ci>
353                     <ci> u </ci>
354                 </apply>
355             </math>
356         </kineticLaw>
357     </reaction>
358     <reaction id="Reaction3" metaid="_175123" reversible="false" sboTerm="SBO:0000176">
359         <listOfReactants>
360             <speciesReference species="z" metaid="_372816"/>
361         </listOfReactants>
362         <listOfProducts>
363             <speciesReference species="u" metaid="_372828"/>
364         </listOfProducts>
365         <kineticLaw metaid="_372840">
366

```

```

367         <math xmlns="http://www.w3.org/1998/Math/MathML">
368             <apply>
369                 <times/>
370                 <ci> k4 </ci>
371                 <ci> z </ci>
372             <apply>
373                 <plus/>
374                 <apply>
375                     <divide/>
376                     <ci> k4prime </ci>
377                     <ci> k4 </ci>
378                 </apply>
379                 <apply>
380                     <power/>
381                     <ci> u </ci>
382                     <cn type="integer"> 2 </cn>
383                 </apply>
384             </apply>
385         </math>
386     </kineticLaw>
387 </reaction>
388 </listOfReactions>
389 </model>
390 </sbml>

```

## E. SED-ML Beispiel

Der folgende SED-ML-Code spezifiziert die Simulation einer Zeitreihe von Modell 1 und die Ausgabe der Ergebnisse als Graph der relativen Konzentrationen  $[M]/[CT]$  und  $[YT]/[CT]$  gegen die Zeit  $t$  (für Details siehe Beispiel 4.4). Der Code wurde mit Hilfe der SED-ML Web Tools ([http://sysbioapps.dyndns.org/SED-ML\\_Web\\_Tools/](http://sysbioapps.dyndns.org/SED-ML_Web_Tools/)) erstellt. Eine ähnliche SED-ML-Datei findet sich im Eintrag der BioModels Database für Modell 1 unter <http://www.ebi.ac.uk/biomodels-main/getAdditionalFile?mid=BIOMD0000000005&fid=48>. Umgebrochene Zeilen sind durch  $\hookrightarrow$  gekennzeichnet.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!-- Written by libSedML v1.1.5227.2088 see http://libsedml.sf.net -->
3 <sedML level="1" version="1" xmlns="http://sed-ml.org/">
4   <listOfSimulations>
5     <uniformTimeCourse id="sim1" initialTime="0"
6                           outputStartTime="0" outputEndTime="100"
7                           numberOfPoints="1000">
8       <algorithm kisaoID="KISA0:0000019" />
9     </uniformTimeCourse>
10  </listOfSimulations>
11  <listOfModels>
12    <model id="model1" language="urn:sedml:language:sbml" source="urn:miriam:biomodels.db:
13      ↪ BIOMD0000000005" />
14  </listOfModels>
15  <listOfTasks>
16    <task id="task1" modelReference="model1" simulationReference="sim1" />
17  </listOfTasks>
18  <listOfDataGenerators>
19    <dataGenerator id="time" name="time">
20      <listOfVariables>
21        <variable id="time" name="time" taskReference="task1" symbol="urn:sedml:symbol:
22          ↪ time" />
23      </listOfVariables>
24      <math xmlns="http://www.w3.org/1998/Math/MathML">
25        <ci> time </ci>
26      </math>
27    </dataGenerator>
28    <dataGenerator id="M_CT" name="M_CT">
29      <listOfVariables>
30        <variable id="M" name="M" taskReference="task1"
31          target="/sbml:sbml/sbml:model/sbml:listOfSpecies/sbml:species[@id='M']"
32          ↪ />
33        <variable id="CT" name="CT" taskReference="task1"
34          target="/sbml:sbml/sbml:model/sbml:listOfSpecies/sbml:species[@id='CT']"
35          ↪ />
36      </listOfVariables>
37      <math xmlns="http://www.w3.org/1998/Math/MathML">
38        <apply>
```

```

35         <divide />
36         <ci> M </ci>
37         <ci> CT </ci>
38     </apply>
39 </math>
40 </dataGenerator>
41 <dataGenerator id="YT_CT" name="YT_CT">
42     <listOfVariables>
43         <variable id="YT" name="YT" taskReference="task1"
44             target="/sbml:sbml/sbml:model/sbml:listOfSpecies/sbml:species[@id='YT']"
↪ />
45         <variable id="CT" name="CT" taskReference="task1"
46             target="/sbml:sbml/sbml:model/sbml:listOfSpecies/sbml:species[@id='CT']"
↪ />
47     </listOfVariables>
48     <math xmlns="http://www.w3.org/1998/Math/MathML">
49         <apply>
50             <divide />
51             <ci> YT </ci>
52             <ci> CT </ci>
53         </apply>
54     </math>
55 </dataGenerator>
56 </listOfDataGenerators>
57 <listOfOutputs>
58     <plot2D id="plot1">
59         <listOfCurves>
60             <curve id="curve_0" logX="false" logY="false" xDataReference="time" yDataReference
↪ ="M_CT" />
61             <curve id="curve_1" logX="false" logY="false" xDataReference="time" yDataReference
↪ ="YT_CT" />
62         </listOfCurves>
63     </plot2D>
64 </listOfOutputs>
65 </sedML>

```



## F. TEDDY: Klassenhierarchie

In diesem Anhang ist die Klassenhierarchie von **TEDDY** (*TErminology for the Description of DYnamics*, COURTOT ET AL. 2011) dargestellt. Die Klassenhierarchie wurde mit Hilfe eines modifizierten Protégé-Plugins (basierend auf <https://code.google.com/p/co-ode-owl-plugins/wiki/TaxonomyCutAndPaste>) erstellt aus der aktuellen Version **rel-2014-04-24** von **TEDDY**. Dargestellt ist die inferierte Klassenhierarchie, die mit Hilfe von **HermiT** (MOTIK ET AL. 2009) gewonnen wurde. Konzepte kommen in dieser Darstellung mehrfach vor, falls sie mehrere Oberkonzept besitzen. Jede Zeile stellt ein **TEDDY**-Konzept mit Bezeichner und zugehöriger verkürzter ID in Klammern dar. Jede ID kann durch Voranstellen von <http://identifiers.org/teddy/> zur vollständigen IRI (*Internationalized Resource Identifier*) erweitert werden (vgl. Abschnitt 6.2.2). Die Anzahl der Striche vor jedem Konzeptbezeichner repräsentiert die Tiefe des Konzeptes im entsprechenden **TEDDY**-Zweig, d.h. die Länge des Pfades von diesem Konzept zum Wurzelkonzept 'TEDDY entity' [**TEDDY\_0000000**].

TEDDY entity (**TEDDY\_0000000**)

- behaviour characteristic (**TEDDY\_0000002**)
- – curve characteristic (**TEDDY\_0000023**)
- – – curve quantitative characteristic (**TEDDY\_0000178**)
- – – – magnitude (**TEDDY\_0000116**)
- – – – – amplitude (**TEDDY\_0000131**)
- – – – – high magnitude (**TEDDY\_0000117**)
- – – – – low magnitude (**TEDDY\_0000118**)
- – – – period (**TEDDY\_0000067**)
- – – curve shape (**TEDDY\_0000015**)
- – – – concave shape (**TEDDY\_0000016**)
- – – – – straight line shape (**TEDDY\_0000022**)
- – – – – convex shape (**TEDDY\_0000021**)
- – – – – straight line shape (**TEDDY\_0000022**)
- – – – sigmoid shape (**TEDDY\_0000032**)
- – – inflexion point (**TEDDY\_0000130**)
- – – limit (**TEDDY\_0000025**)
- – – – asymptotic limit (**TEDDY\_0000026**)
- – – – – asymptotic lower limit (**TEDDY\_0000028**)
- – – – – asymptotic upper limit (**TEDDY\_0000027**)
- – – – infinite limit (**TEDDY\_0000030**)
- – – – – minus infinity limit (**TEDDY\_0000031**)
- – – – – plus infinity limit (**TEDDY\_0000029**)
- – – monotonicity characteristic (**TEDDY\_0000144**)
- – – – monotonicity (**TEDDY\_0000004**)

```

----- strict monotonicity (TEDDY_0000007)
----- strictly decreasing (TEDDY_0000009)
----- linear decreasing (TEDDY_0000020)
----- strictly increasing (TEDDY_0000008)
----- linear increasing (TEDDY_0000019)
----- non-monotonic (TEDDY_0000005)
----- oscillation (TEDDY_0000006)
----- chaotic oscillation (TEDDY_0000061)
----- damped oscillation (TEDDY_0000063)
----- periodic oscillation (TEDDY_0000066)
----- mixed-mode oscillation (TEDDY_0000065)
----- single-periodic oscillation (TEDDY_0000064)
----- quasi-periodic oscillation (TEDDY_0000112)
----- single turnaround (TEDDY_0000010)
---- rate of change (TEDDY_0000012)
---- exponential rate of change (TEDDY_0000014)
---- polynomial rate of change (TEDDY_0000018)
---- linear rate of change (TEDDY_0000013)
---- zero rate of change (TEDDY_0000017)
-- stability characteristic (TEDDY_0000148)
-- half-stability (TEDDY_0000134)
-- hyperbolicity (TEDDY_0000157)
-- Lyapunov stability (TEDDY_0000133)
---- asymptotic stability (TEDDY_0000097)
---- global stability (TEDDY_0000136)
---- only local stability (TEDDY_0000135)
---- neutral stability (TEDDY_0000098)
---- uniform stability (TEDDY_0000150)
-- Poincaré stability (TEDDY_0000149)
-- structural stability (TEDDY_0000167)
-- behaviour diversification (TEDDY_0000132)
-- bifurcation (TEDDY_0000053)
-- global bifurcation (TEDDY_0000147)
-- bifurcation of limit cycle (TEDDY_0000070)
-- blue sky catastrophe (TEDDY_0000177)
-- Neimark-Sacker bifurcation of limit cycle (TEDDY_0000078)
-- period-doubling bifurcation of limit cycle (TEDDY_0000077)
-- saddle-node bifurcation of limit cycles (TEDDY_0000076)
-- heteroclinic bifurcation (TEDDY_0000153)
-- homoclinic bifurcation (TEDDY_0000154)
-- saddle-node homoclinic bifurcation (TEDDY_0000119)
-- local bifurcation (TEDDY_0000069)
-- Hopf bifurcation (TEDDY_0000072)
-- degenerate Hopf bifurcation (TEDDY_0000120)
-- subcritical Hopf bifurcation (TEDDY_0000073)
-- supercritical Hopf bifurcation (TEDDY_0000074)
-- saddle-node on invariant circle bifurcation (TEDDY_0000075)

```

---

```

-- -- -- zero-eigenvalue bifurcation (TEDDY_0000123)
-- -- -- pitchfork bifurcation (TEDDY_0000122)
-- -- -- -- subcritical pitchfork bifurcation (TEDDY_0000124)
-- -- -- -- supercritical pitchfork bifurcation (TEDDY_0000125)
-- -- -- saddle-node bifurcation (TEDDY_0000071)
-- -- -- transcritical bifurcation (TEDDY_0000121)
-- -- bursting (TEDDY_0000055)
-- -- perturbation response (TEDDY_0000108)
-- -- bistable perturbation response (TEDDY_0000110)
-- -- excitability (TEDDY_0000179)
-- -- monostable perturbation response (TEDDY_0000109)
-- -- multistable perturbation response (TEDDY_0000111)
-- functional motif (TEDDY_0000003)
-- feed-forward loop (TEDDY_0000036)
-- -- multi-output feed-forward loop (TEDDY_0000165)
-- -- three-node feed-forward loop (TEDDY_0000037)
-- -- -- coherent three-node feed-forward loop (TEDDY_0000038)
-- -- -- -- type-1 coherent three-node feed-forward loop (TEDDY_0000039)
-- -- -- -- type-2 coherent three-node feed-forward loop (TEDDY_0000040)
-- -- -- -- type-3 coherent three-node feed-forward loop (TEDDY_0000041)
-- -- -- -- type-4 coherent three-node feed-forward loop (TEDDY_0000042)
-- -- -- incoherent three-node feed-forward loop (TEDDY_0000043)
-- -- -- -- type-1 incoherent three-node feed-forward loop (TEDDY_0000044)
-- -- -- -- type-2 incoherent three-node feed-forward loop (TEDDY_0000045)
-- -- -- -- type-3 incoherent three-node feed-forward loop (TEDDY_0000046)
-- -- -- -- type-4 incoherent three-node feed-forward loop (TEDDY_0000047)
-- -- feedback loop (TEDDY_0000033)
-- -- -- negative feedback loop (TEDDY_0000034)
-- -- -- positive feedback loop (TEDDY_0000035)
-- -- -- -- two-node positive feedback loop (TEDDY_0000162)
-- -- -- -- double-negative feedback loop (TEDDY_0000164)
-- -- -- -- double-positive feedback loop (TEDDY_0000163)
-- -- integrator (TEDDY_0000079)
-- -- switch (TEDDY_0000056)
-- -- -- toggle switch (TEDDY_0000166)
-- temporal behaviour (TEDDY_0000083)
-- fixed point (TEDDY_0000086)
-- -- degenerate node (TEDDY_0000093)
-- -- -- stable degenerate node (TEDDY_0000141)
-- -- -- unstable degenerate node (TEDDY_0000142)
-- -- half-stable fixed point (TEDDY_0000104)
-- -- -- saddle (TEDDY_0000089)
-- -- -- Lyapunov stable fixed point (TEDDY_0000095)
-- -- -- neutrally stable fixed point (TEDDY_0000115)
-- -- -- center (TEDDY_0000088)
-- -- -- stable degenerate node (TEDDY_0000141)
-- -- -- stable node (TEDDY_0000137)

```

```

-- stable spiral (TEDDY_0000126)
-- stable star (TEDDY_0000139)
-- node (TEDDY_0000087)
-- stable node (TEDDY_0000137)
-- unstable node (TEDDY_0000138)
-- non-isolated fixed point (TEDDY_0000092)
-- spiral (TEDDY_0000091)
-- stable spiral (TEDDY_0000126)
-- unstable spiral (TEDDY_0000127)
-- star (TEDDY_0000090)
-- stable star (TEDDY_0000139)
-- unstable star (TEDDY_0000140)
-- unstable fixed point (TEDDY_0000129)
-- saddle (TEDDY_0000089)
-- unstable degenerate node (TEDDY_0000142)
-- unstable node (TEDDY_0000138)
-- unstable spiral (TEDDY_0000127)
-- unstable star (TEDDY_0000140)
-- limit behaviour (TEDDY_0000085)
-- attractor (TEDDY_0000094)
-- regular attractor (TEDDY_0000175)
-- stable limit cycle (TEDDY_0000114)
-- stable node (TEDDY_0000137)
-- stable spiral (TEDDY_0000126)
-- stable star (TEDDY_0000139)
-- strange attractor (TEDDY_0000054)
-- repeller (TEDDY_0000102)
-- unstable limit cycle (TEDDY_0000128)
-- unstable node (TEDDY_0000138)
-- unstable spiral (TEDDY_0000127)
-- unstable star (TEDDY_0000140)
-- non-periodic orbit (TEDDY_0000143)
-- asymptotic behaviour (TEDDY_0000146)
-- heteroclinic orbit (TEDDY_0000151)
-- heteroclinic saddle connection (TEDDY_0000101)
-- homoclinic orbit (TEDDY_0000152)
-- homoclinic saddle connection (TEDDY_0000100)
-- saddle connection (TEDDY_0000099)
-- heteroclinic saddle connection (TEDDY_0000101)
-- homoclinic saddle connection (TEDDY_0000100)
-- strange attractor (TEDDY_0000054)
-- periodic orbit (TEDDY_0000050)
-- limit cycle (TEDDY_0000051)
-- half-stable limit cycle (TEDDY_0000145)
-- stable limit cycle (TEDDY_0000114)
-- unstable limit cycle (TEDDY_0000128)
-- non-isolated cycle (TEDDY_0000106)

```

# Ehrenwörtliche Erklärung

Hiermit erkläre ich,

- dass mir die geltende Promotionsordnung der Fakultät für Mathematik und Informatik Jena bekannt ist,
- dass ich die Dissertation selbst angefertigt habe,
- dass ich keine Textabschnitte oder Ergebnisse eines Dritten oder eigener Prüfungsarbeiten ohne Kennzeichnung übernommen habe,
- dass ich alle benutzten Hilfsmittel, persönlichen Mitteilungen und Quellen angegeben habe,
- dass ich keine Hilfe eines Promotionsberaters in Anspruch genommen habe,
- dass Dritte weder unmittelbar noch mittelbar geldwerte Leistungen von mir für Arbeiten erhalten haben, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen,
- dass ich die Dissertation noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht habe und
- dass ich nicht die gleiche, eine in wesentlichen Teilen ähnliche oder eine andere Abhandlung bei einer anderen Hochschule als Dissertation eingereicht habe.

Jena, den 22. Oktober 2015

.....